MASTER IN COMPUTER SCIENCE

Visualization and exploration tool for highly multi-dimensional medical data

Martin, Amélie

Award date:
2010

Awarding institution:
University of Namur

University of Namur
Faculty of Computer Science

2009-2010

Visualization and exploration tool for highly multi-dimensional medical data

Martin Amélie

A thesis presented to the Faculty of Computer Science
in partial fulfillment of the requirements for the Degree
Master of Computer Science

Abstract

Nowadays, data sets are not only bigger in their size but also in their dimensionality. Interactive visualization of highly multi-dimensional data sets is crucial to present and discover the insight, pattern and (ir)regularities of information. The classical methods are not usually efficient to handle medical data sets with so many attributes. This thesis presents a tool to visualize and explore a data set that contains several thousands of attributes. This tool works in two phases. The first one is the reduction of the amount of attributes, through the construction of data groups, transformed in turn into interval data. The second phase visualize the low multi-dimensional data using 3-dimensional multiple scatter plot. The system also provides interactive exploration so that analysts can view, interact and manipulate the processed data as well as the information in its original form.

Résumé

De nos jours, la taille et la complexité des jeux de données utilisés par les techniques du data mining augmentent continuellement. La visualisation interactive de jeux de données contenant un très grand nombre de variables est crucial pour présenter un aperçu et découvrir des tendances et des (ir)régularités dans les données. Les méthodes classiques ne sont pas assez efficaces pour gérer des jeux de données médicales contenant autant de variables. Ce mémoire présente un outil qui permet de visualiser et d'explorer un jeu de données contenant plusieurs miliers de variables. Cet outil fonctionne en deux phases. La première consiste en la réduction du nombre de variables par une construction de groupes de variables à leur tour transformés en données intervalles. La deuxième phase permet de visualiser les données intervalles en utilisant des visualisation en trois dimensions. L'outil fournit aussi la possibilité d'explorer interactivement les données pour que l'analyste puisse visualiser, intéragir et manipuler les données après transformation mais également dans leur forme originale.

Acknowledgments

I would like to thank several people who helped me during the elaboration of this thesis.

First of all, I would like to thank Professor Monique Noirhomme for the opportunity she gave me to realize this thesis under her supervision and for her help during the writing.

I would like to express my gratitude to Quang Vinh Nguyen and Simeon Simoff who gave me the opportunity to realize my internship at the Western University of Sydney but also for their help, support and ideas during this internship and the writing of this thesis.

I would also like to thank Daniel Catchpoole from the Children's Hospital at Westmead for the data set on which this work is based, for the information about it and also for the feedback provided over the tool.

I will finish these acknowledgements by thanking the people who read this thesis and gave me some corrections and advices about it, especially Alain Mordant.

Contents

1 Introduction **1**

2 State of the art **5**
2.1 Information visualization and visual analytics 5
2.2 Taxonomy of visualization techniques 7
2.2.1 The data type 7
2.2.2 The visualization techniques 8
2.2.3 The interaction and distorsion techniques 12
2.3 Dimensional reduction methods 14
2.3.1 Principal component analysis (PCA) 14
2.3.2 Multidimensional scaling (MDS) 15
2.3.3 Locally linear embedding (LLE) 16
2.3.4 Laplacian eigenmap 16
2.3.5 Comparisons of the reduction methods 17

3 Treatment of the data set **21**
3.1 Construction of groups of attributes 21
3.1.1 Basic idea for the construction of the groups 22
3.1.2 First parameter : the mean 22
3.1.3 Second parameter : the coefficient of variation 23
3.1.4 The scope of the data as a reference for the mean 23
3.1.5 The choice of the values for p_m and p_{CV} 24
3.1.6 Optimization : The use of the correlation in a group 24
3.2 Use of the groups : construction of interval data 27
3.2.1 General idea behind the construction 27
3.2.2 The symbolic data 27
3.2.3 The construction of interval data from the groups 29
3.3 Principal component analysis for interval data 31
3.4 The general algorithm 34
3.4.1 First construction of the group 34
3.4.2 Optimization of the groups 35
3.4.3 Construction of the interval data 39
3.5 Results for the gene data set 40
3.5.1 The parameters for the mean and the coefficient of variation 40
3.5.2 Number of groups before the optimization 40

3.5.3 Number of groups after the optimization 41
3.5.4 Construction of the symbolic data 42
3.5.5 Principal component analysis results 42

4 Visualization **45**
4.1 Multiple scatter plot 45
4.1.1 General idea 45
4.1.2 Representation of symbolic data 46
4.1.3 The representation of the interval value on the scatter plots . 53
4.1.4 Application to the principal components built on the gene data set 56
4.1.5 Applicable modifications on the multiple scatter plot 56
4.1.6 Discussion 56
4.2 Level circles representation 58
4.2.1 General idea 58
4.2.2 The choice of the represented distance and normalization of these distances 59
4.2.3 The choice of the number of attributes by level 59
4.2.4 Representation of the levels on the reference axis 59
4.2.5 Options applicable on this representation 60
4.2.6 Discussion 61

5 Exploration **63**
5.1 The principles of the exploration 63
5.2 Visualizations that help to choose which part to explore 65
5.2.1 Scatter plot for the groups 65
5.2.2 Information on the groups for each principal component . . . 66
5.2.3 Information on all the principal components 67
5.3 The choice of the next step of the exploration 67
5.3.1 The selection of a principal component 68
5.3.2 The selection of a group 68
5.3.3 The selection of some attributes 69

6 An example of exploration **71**
6.1 First step of the exploration 71
6.1.1 Exploration of the multiple scatter plot 72
6.1.2 Focus on some items and distances between them 74
6.1.3 Focus on one item on all the scatter plots 75
6.1.4 Getting some information on the attributes 77
6.1.5 Selection of a principal component 78
6.1.6 Selection of a group 78
6.1.7 Selection of a subset of attributes 80
6.2 Second step of the exploration 82

7 Conclusion **83**
7.1 Discussion 83
7.2 Future work 84

Annexes **I**

8 Manual for the tool **I**

List of Figures

2.1 The three axes in the taxonomy of Grinstein, Keim and Ward [14] . 7
2.2 Scatter plot matrix for the famous iris data set [28] 9
2.3 Example of parallel coordinates [15] 9
2.4 Examples of Chernoff faces [4] 10
2.5 Example of a color icon [7] 10
2.6 Examples of the circle segments [17] 11
2.7 Example of recursive pattern [19] 11
2.8 Example dimentional stacking with four attributes [30] 12
2.9 Typical data sets tested [12] 18

3.1 Different values of the coefficient of correlation [29] 25
3.2 An item on which two interval variables are measured 31
3.3 Representation of the groups 41

4.1 Multiple scatter plot 46
4.2 Representation of an item described by two interval variables 47
4.3 An example of 2D zoom star [25] 48
4.4 Histogram of a modal attribute [25] 48
4.5 Comparisons of two items by superimposition [25] 49
4.6 An example of 3D zoom star [25] 49
4.7 An example of lattice [9] 50
4.8 Representation of the clusters by a circle and a square [25] 51
4.9 The same representation with a bar diagram [25] 51
4.10 Representation of each clusters by a zoom star [25] 52
4.11 Representation of each clusters by a bar diagram [25] 52
4.12 VPLOT representation [25] 53
4.13 Representation of the four items with the size and the color of the spheres 55
4.14 Representation of the two first reference axes on the level circles representation 60
4.15 Visualization of a level 61

5.1 Scatter plot for the groups 65
5.2 Influence of the groups on one principal component 66
5.3 Percentage of variation explained by each principal component . . . 67

6.1 First multiple scatter plot 71
6.2 First scatter plot from the multiple scatter plot representing the two first principal components 72
6.3 Selection of the two items we will compare 73
6.4 Exact values for the two selected items 73
6.5 Distances representation for the 12000 first attributes 74
6.6 Distances representation for the 10280 last attributes 75
6.7 Representation of the first level of the first basis 76
6.8 Representation of the sixth item on all the scatter plots 76
6.9 The variation explained by each principal components 77
6.10 Influence of the groups on the first principal component 77
6.11 Multiple scatter plot obtained once we select the first principal component 78
6.12 Multiple scatter plot obtained once we select the group 39 79
6.13 Multiple scatter plot obtained once we select the group 133 80
6.14 Scatter plot of the groups 80
6.15 Selection of 2 attributes 81
6.16 Selection of 3 attributes 81

Chapter 1

Introduction

With the development of the different possibilities to collect data and the decrease of the materials storage cost, we find nowadays bigger and bigger data sets. Now, data can be collected more easily than before (a lot of information is available from the internet, we can realize some polls more easily and in a lot of different ways). We want to use these data to extract some useful information or to explain some phenomena. The more characteristics (represented by some attributes) we have on a given subject, the more information we can have on it.

But the data itself, as we can find them in some files, are not always useful : we do not want to read all the values but rather extract some information from them. Indeed, reading and understanding a large amount of data is a difficult and time-consuming task. A human can only focus on a limited number of data at a time. Miller proved that we can only work on approximately seven elements simultaneously [9]. This is especially the goal of developing visualization methods : such methods allow to distinguish some trends, notice some abnormal values and some other characteristics of the data.

The problem with huge data sets is that a person can not easily see some general information about it if we do not treat them. There are too many attributes and too many items on which measurements have been taken. A viewer can then not, on a simple representation, get an idea of what is behind the data. But when people are making such big data sets they hope to discover some information in it. That is the reason why the area of data visualization became so important. There is a lot of possibilities when working on simple or medium data sets but it is more difficult to work on huge data sets. Indeed, the methods developed for small or medium data sets can not always be adapted to be used in the case of data sets containing thousands of attributes or items.

In the family of huge data sets there is a type of data sets that contain medical data, especially gene measurements. In such data sets, we can find several thousands of attributes. This particular type of data set will be our main subject here.

We will detail a solution to visualize and explore a gene data set containing 22.280 attributes, each one corresponding to a particular gene. More precisely, each genes is represented by one oligonucleotide (a short polymer of two to twenty nucleotides [3]) or by several to gain coverage of particular long genes. The attributes are the quantification of the oligonucleotides and they are then quantitative variables. The measures were taken on 196 different patients. This data set belongs to Kids Research Institute of the Children's Hospital at Westmead (CHW, Sydney, Australia) and they are responsible for its generation. Samples were obtained from the Children's Hospital Tumour Bank. These samples were collected according to institutional guideline and with the permission of the CHW Research ethics Committee and the Tumour Bank Committee.

More precisely, the data set concerns patients who have cancer or leukemia. The measures of the various genes were taken on some patients from the Westmead hospital in Sydney while other measures were taken in Washington. The data set also include some measures from people who were not sick. The name of the items reflects their origins (Westmead, Washington or healthy people).

In this data set, the data were previously normalized. We will not detail the normalization procedure that was used. This procedure was quite experimental (they had to adapt the procedure because they had problems to handle such big data set). This normalization has been realized before we began to work on the data set by the Kids Research Institute.

The idea here was to develop a general solution just by taking into account the number of attributes, and trying to find a way to see some type of information. If we wanted to focus on the meaning of the data, we would have to understand all the biological concepts behind the data set. Furthermore, we would also have to study all the interesting biological concepts that are necessary to develop a visualization reflecting the possible results for these concepts.

What we want to do is to find a way, using some statistical methods and visualizations, to visualize and explore these data, especially the values for the genes. The project here was to allow a user to have information about the data with the particularity that this user has absolutely no idea of what he can discover in these data. Furthermore, the user is not always a computer scientist or a mathematician, so, the solution had to be understandable by people who do not have any knowledge in these domains. So, it has to be as simple as possible to understand and use by demanding the less possible knowledge in mathematics and computer science.

The solution found here is a tool which allows the user to explore the genes values in a progressive way. From a general overview of the data, he can find some information that help him to choose what part of the data he wants to focus on in the next step. By selecting a part of the data, the user selects a subset of attributes. Indeed, with this tool, we focus on the attributes side of the data set, not on the items. At

each step, he can visualize the data and some characteristics with some graphics to obtain some general information about the selected attributes. These graphics help him to determine what genes are the most important and on which he could focus on during the following steps.

In the second chapter we will first define the general framework in which the solution is located. Then, we will study one of the existing taxonomy of the visualization techniques with some examples. From these examples, we will see the importance to use a dimensional reduction method to help us to visualize a high multi-dimensional data set. We will make a short review of this kind of methods.

From the third chapter, we will explain the solution that we found for the gene data set. In the third chapter, we will begin by a description of the treatment that we apply on the data before the visualization. Indeed, it is not possible to work directly on all the attributes and we need to find a way to decrease the number of attributes before starting the visualization.

In the fourth chapter, we will explain the different visualization methods that we will use in the development of the tool.

In the fifth chapter, we will see the general principle of the exploration of the data set. We will examine the different options in the exploration and what these choices imply.

In the sixth chapter we will see an example of the exploration to have an idea of how the visualizations and the principles of the interaction can be combined in order to explore the entire data set.

We will then finish with a conclusion and see what could still be done to improve the tool developed here.

Chapter 2

State of the art

In this chapter, we will introduce the main concepts that we will use to present a solution for the visualization and exploration of the genes data set.

First, we will define the framework in which the solution developed here take place : the visual analytics. After this, we will present one of the possible taxonomies for the visualization techniques. We will then present some visualization methods. The last part of this chapter will be a review of some dimensional reduction techniques. Indeed this kind of methods is very important when we want to study a data set with so many attributes, as in the gene data set.

2.1 Information visualization and visual analytics

All along the different chapters, we will address the data visualization. The data visualization can be seen as a narrower domain of the general domain of **information visualization**. Information visualization can be defined as "the visual representation of large-scale collections of non-numerical information, such as files and lines of code in software systems, libraries and bibliographic databases, networks of relations on the internet, and so forth" ([13]). We will introduce a taxonomy of this area in the next section.

The **data visualization** can be defined as "the science of visual representation of data" ([13]) where the data are defined as "information which has been abstracted in some schematic form, including attributes or variables for the units of information" ([13]). These areas are just visualizations and that is what distinguish them from the area of visual analytics.

Visual analytics is defined as "the science of analytical reasoning facilitated by interactive visualization for an effective understanding" ([13]). The areas of visual analytics and information visualization seem to be very similar. There are indeed some overlays between the two areas, but there is still a big difference : the information visualization does not deal with an analysis task and it does not always use

some data analysis algorithms.

As we will see later when we will describe the solution for the gene data set, this solution enters in the area of visual analytics. We do not just visualize some information (even if we will do that too) but we also analyze the data before visualize them. We will then now take a closer look at this area.

We already notice in the introduction that we find more and more data and there is a danger with such amount of data : **the information overload problem** defined by Daniel Kheim [8] as "the danger of getting lost in data which may be irrelevant to the current task at hand or processed or presented in an inappropriate way". The visual analytics try to turn this danger into an opportunity. Indeed it tries to give an effective understanding, a capacity of reasoning and decision making on the basis of very large and complex data sets.

As described by Daniel Kheim in the same article [8], the aim of visual analytics is to develop some tools and techniques that will enable the people to :

- synthesize information and derive insight from massive, dynamic, ambiguous and often conflicting data
- detect the expected and discover the unexpected
- provide timely, defensible and understandable assessments
- communicate assessments effectively for action.

To achieve these goals, the visual analytics use both strengths of humans and electronic data processing. Indeed, the data will be processed but the user will have to choose the direction of the analysis. That is where the interactive part of the area is expressed : the data will be treated and presented to the user. He will then be able to interact with these data in order to discover some information before choosing the direction for the analysis.

Now that we saw the main characteristics of the visual analytics, we will come back to the area of information visualization and study a taxonomy for this area.

2.2 Taxonomy of visualization techniques

As for every scientific discipline, research have been conducted to build a taxonomy for the information visualization. Since 1990, there has been different proposals. We will not describe each of this taxonomies, a complete description can be found in the thesis of Benoît Otjacques [24]. Here, we will retain the one proposed by Grinstein, Keim and Ward between 2001 and 2004 [14].

Grinstein, Keim and Ward proposed a taxonomy based on three different axes :

- the data type we want to visualize
- the visualization technique
- the interaction and distortion technique.

Keim defines these three axes as orthogonal which means that any visualization technique can be used in conjunction with any interaction technique as well as any distortion style for any data type.

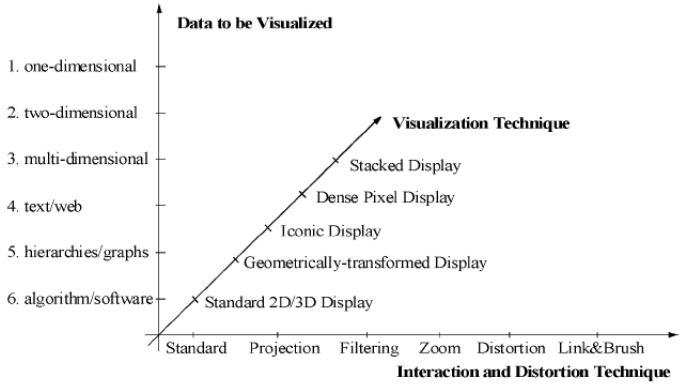


Figure 2.1: The three axes in the taxonomy of Grinstein, Keim and Ward [14]

In the following sections, we will describe these three different axes.

2.2.1 The data type

In their taxonomy, Grinstein, Keim and Ward distinguish six different data types we can visualize :

- **One-dimensional data** : In this case, data usually have one dense dimension. An example of such data is temporal data such as time series of stock prices.

- **Two-dimensional data** : Here, we only have two distinct dimensions. As an example, we can consider geographical data where the two dimensions are the longitude and the latitude.
- **Multidimensional data** : In this case, the data have more than three dimensions. An example is the relational databases which often have tens to hundreds of columns and each column represents an attribute and thus one dimension. Here, the data do not allow a simple visualization as 2-dimensional or again 3-dimensional plots. We then need more sophisticated visualization techniques.
- **Text, hypertext** : Every data can not be described in terms of a dimensionality. As an example, we can have text, hypertext or again web page contents. As the data can not easily be described by numbers, a transformation has to be applied on the data before starting to use a visualization technique (an example of such a transformation is the word counting).
- **Hierarchies, graphs** : Sometimes, the data have some relationship to other pieces of information. Graphs and hierarchies are used to represent these relationships. Some examples are the file structure of a hard disk, the shopping behaviour of a person or the hyperlinks in the World Wide Web.
- **Algorithms, software** : In this case, the aim of the visualization is to support software development by helping to understand algorithms by representing the structure of source code. This can also be done to support the programmer in debugging the code by visualizing errors.

2.2.2 The visualization techniques

For the visualization techniques, they proposed five different categories :

- **Standard 2D/3D display** : This category includes all the classical visualization techniques as the x-y or x-y-z plots, the bar charts, ...
- **Geometrically-transformed display** : This category of visualization techniques aims at visualizing some geometrical transformations or some projections of the data. Some examples are the scatter plot matrix or the parallel coordinates.

  The *scatter plot matrix* represents a matrix of all the possible scatter plots we can build on the data. A scatter plot uses a cartesian coordinates to display the values of some attributes (two or three depending on the number of axes we choose for the scatter plot). The scatter plot thus corresponds to a projection on a space of two or three dimensions. On the scatter plot matrix, we will thus have all the combinations of the possible projections.

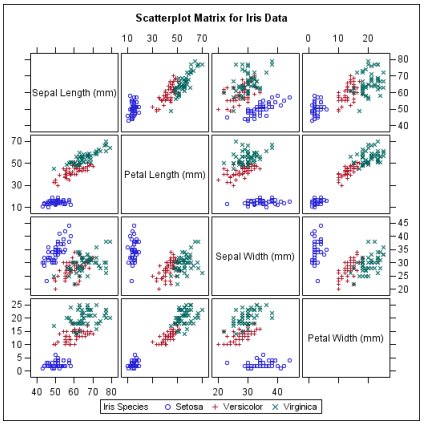


Figure 2.2: Scatter plot matrix for the famous iris data set [28]

The geometrical-transformed display also include the *parallel coordinates*. In this representation, if we have n attributes, we will draw n parallel lines. A n-dimensional point will be represented by a polyline with vertices on each parallel line. The position of a vertex on a parallel line representing an attribute will correspond to the value of this attribute for that point.

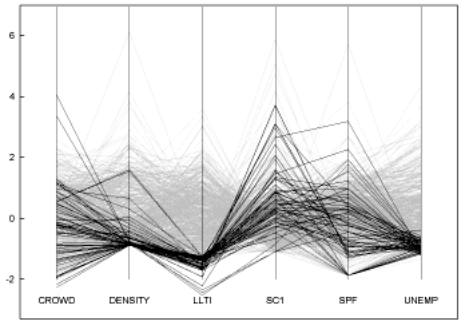


Figure 2.3: Example of parallel coordinates [15]

- **Iconic display** : These techniques try to map the attributes values of a multi-dimensional data item to the features of an icon. We can cite as examples the Chernoff faces or the color icons.

  In the *Chernoff faces*, an attribute is represented by a characteristic of a face (for example the face width, the level of the ears, the length or the curvature of the mouth,...). The shape and the size of each characteristic vary according to the values of its variables.

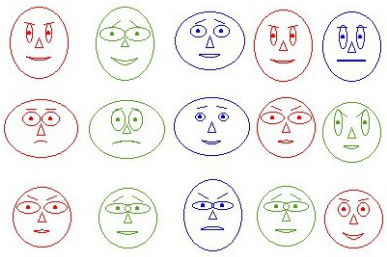


Figure 2.4: Examples of Chernoff faces [4]

Another example is the *color icon*. Here, we have a square or a hexagon which is divided into several areas. Each area is in a different color and represents one parameter.

Figure 2.5: Example of a color icon [7]

- **Dense pixel display** : The aim here is to map each dimension to a colored pixel and group the pixels belonging to each dimension into adjacent areas. One pixel represents one data value and thus, these techniques allow us to visualize the largest amount of data possible on current displays. Some examples of these methods are the recursive pattern technique or the circle segments technique.

  In the *circle segments* method, we represent each attribute by a segment of a circle. If we have eight attributes, the circle will be divided into eight segments. After that, all the data items are organized in a back and forth manner along a line, called draw line, which is orthogonal to the line that divide the two border limits of the segments. After that, we draw each pixel starting in the center of the circle from one border line of the segment to the other.

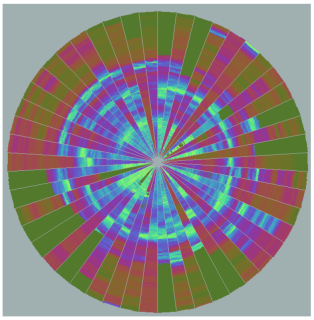


Figure 2.6: Examples of the circle segments [17]

Another dense pixel display is the *Recursive Pattern Technique*. Here, each attribute is presented in a separate subwindow. In this window, each value of the attribute for each item is represented by a colored pixel and the color represents the value of this attribute.

Figure 2.7: Example of recursive pattern [19]

- **Stacked display** : These techniques visualize the data in a hierarchical fashion. An example of such technique is the Dimensional Stacking.

  In the *dimensional stacking*, we work on two dimensions and each axis, representing an attribute, is break into some divisions (one division for each possible value for this attribute). By doing this, we obtain some new areas. Each of these areas is considered as a new system of axes, which represent two new attributes that we can divide too.

  On this example, four attributes are represented. The first two attributes are represented on the x and y axes. On this figure, both attributes have four possible values, they are thus divided into four divisions (corresponding to the red lines). To represent the two other attributes, we do the same division inside each rectangle defined by the first division. We will represent a third attribute horizontally inside a rectangle and a fourth one vertically. Both have also four possible values. The second division correspond to the green lines.

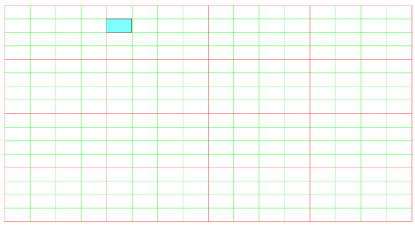


Figure 2.8: Example dimentional stacking with four attributes [30]

2.2.3 The interaction and distorsion techniques

The last axis concerns the interaction and distortion techniques. These techniques aim at realizing an effective data exploration. The *interaction techniques* allows to directly interact with the visualizations and dynamically change it according to the objectives of the exploration. They also allow relating and combining different independent visualizations. The *distortion techniques* provide means to focus on details while preserving an overview of the data : some parts of the data are shown with a high level of detail while the other parts are shown with a lower level of details.

Keim also distinguishes the dynamic and interactive techniques. In the dynamic techniques, the modifications to the visualization are made automatically and in the interactive techniques, they are made manually by the user of the visualization.

They distinguish six different categories of techniques for the interaction and the distortion :

- **Standard** : This category contains all the classical methods.
- **Projection** : The idea is to dynamically change the projections in order to explore a multidimensional data set.
- **Filtering** : This method is used in the exploration of large data sets and it is an interactive method. The aim is to partition the data set into segments and focus on interesting subsets. In order to do this, we can directly select a desired subset (browsing) or we can specify the properties of the desired subset (querying).
- **Zoom** : This is an interactive technique. The general aim is to present the data in a highly compressed form to give an overview of the data. To this overview, we add the possibility to realize different displays of the data on different resolutions. It is not only the idea to show an object greater but it is also the idea to change the representation to show more details.
- **Distortion** : It is an interactive technique and it allows to have a more detailed representation of a part of the data, while keeping an overview of the other parts.

- **Link & Brush** : The idea is to combine different visualization methods in order to overcome the weaknesses of each of the individual methods.

In this section we saw some examples of different visualization techniques we can use. These methods may sometimes be usable to represent a lot of items (as in the recursive pattern technique) but they are never usable to represent a lot of attributes. If we want to be able to represent a data set that containing a large number of attributes we can apply a dimensional reduction method, which will be the subject of the next section.

2.3 Dimensional reduction methods

When we try to visualize the data contained in the genes data set, the first big problem we are facing is the number of attributes. Indeed, on the data set on which we are working, there are approximately twenty two thousands attributes, one for each gene measured on each patient. As we saw in the previous section, the classical visualization methods are not suitable for so many attributes. For example, if we want to realize a scatter plot matrix for this data set, the number of scatter plots is really too big to allow a good understanding of the data.

If we want to visualize such data, we thus need to reduce the number of attributes, in order to use these visualization methods. If we want to keep all the attributes, because they all have a meaning in the data set, we need to find some new attributes that will summarize the information contained in the original attributes.

In this section, we will examine some of the methods that allow us to do this. Each of these methods tries to decrease the number of attributes by keeping as many information as possible from the original data set.

One general remark about the methods that we will explain afterwards is that they all take as input the original data (all data are represented as points in a high-dimensional space) while the output remains all these points but in a lower-dimensional space.

We can formalize the problem of dimensional reduction in the following way :

We assume that the original data consists in the measurement of n attributes made on a certain number of items. These attributes are denoted by x_i $(i = 1, ..., n)$. The aim of a dimensional reduction method is to find l new attributes y_j. If we want to have a certain visualization for the data set, most of the time, we want l not to be greater than three, but it depends on the visualization method we want to use. The new attributes y_j had to reflect the essential properties of the original attributes x_i but clearly, by reducing the number of attributes, it is nearly impossible to keep all the information included in the original data set.

We are now going to see some examples of this type of methods. Of course, this is not an exhaustive list because there are a lot of different possibilities and each method exists in different versions.

2.3.1 Principal component analysis (PCA)

The aim of the principal component analysis is to build new variables, called principal components, as a linear combination of the original variables. The number of principal components is smaller than the number of original attributes. These principal components are built in a way that they keep the maximum percentage of

variance of the original data.

The principal components are easily found by solving an eigenvalue problem on the covariance matrix. We have to find the solutions to the problem :

$$Cv = \lambda v$$

where C is the covariance matrix of the original data (if we have p original attributes, C will be of dimension $p * p$), v is an eigenvector and λ is the associate eigenvalue. This problem has several solutions and the new attributes y_j will be built from the eigenvector v_j that we find by solving this problem :

$$y_j = v_j X$$

where X is the vector of the original attributes $(X_1, ..., X_p)$.

The principal component will be a linear combination of the original attributes and the coefficients of these original attributes will be given by the eigenvectors.

To keep the biggest possible percentage of variance from the original data, we will keep the l eigenvectors corresponding to the l biggest eigenvalues.

The version we explained here is the basic version of principal components analysis but there is a lot of different versions.

2.3.2 Multidimensional scaling (MDS)

The general aim of the multidimensional scaling is to find a configuration of points in a space that preserves the pairwise distances matrix as well as possible. There are different versions of the multidimensional scaling.

The simplest is the **classical scaling** where the solutions for the new configuration are found by solving an eigenvalue problem on the matrix containing the squared distances between the points, exactly as in PCA.

Another version of the MDS is the **metric MDS** in which we associate a loss function representing the quality of the representation instead of just a distances matrix. This loss function has to be minimized :

$$E = \sum (d_{i,j} - d(y_i, y_j))^2$$

where $d_{i,j}$ is the distance between the points i and j in the original space and $d(y_i, y_j)$ is the distance between the representation $y_{i(j)}$ of the point $i(j)$ in the lower dimensional space.

It is the result of the optimization of this loss function that gives us the transformation that we need to apply to the points to obtain their representation in the new

space.

We can find again different versions of the MDS by modifying this loss function.

These 2 methods (PCA and MDS) are linear methods even if there are some non linear versions of MDS. The two next methods that we will see are non linear methods.

2.3.3 Locally linear embedding (LLE)

In this method, instead of using the criteria of preserving the pairwise distances, a point is reconstructed by its neighbours. This is based on the idea that a point and its neighbors lies on or close to a lower-dimensional subspace. The method tries to approximate this subspace to obtain a lower-dimensional representation.

The LLE algorithm works in two phases :

- Each point from the original data can be reconstructed thanks to its neighbors. To express this fact, we use a matrix W where W_{ij} represents the contribution of the j-th data point, X_j to the reconstruction of the item i, X_i. To have a good formalization of the problem, we had to say that a point is only reconstructed by its neighbors which means that $W_{ij} = 0$ if X_j is not in the set of neighbors of X_i. An additional constraint is that the sum of a row had to be equal to one : $\sum_j W_{ij} = 1$. To find the optimal matrix of weights, we need to optimize the cost function of the representation :

$$\epsilon(X) = \sum_i |x_i - \sum_j W_{ij} x_j|^2$$

- In the second step of the algorithm, we search the optimal representation Y_i for the item X_i by choosing this representation as the minimum of the following function :

$$\epsilon(X) = \sum_i |Y_i - \sum_j W_{ij} Y_j|^2$$

  but this time, the matrix W is known.

2.3.4 Laplacian eigenmap

In this method, the idea is related to the one of the LLE method, in the sense that we work with the neighbors of the points. But in this case, we need to build a neighbourhood graph which defines the similarities that we would like to see in the final visualization. The Laplacian eigenmap then try to represent the structure shown on this graph by optimizing a certain cost function taking into account the weight associated to the edges. The idea is to try to let the points in the representation close one from each other if they are connected in the graph.

The algorithm works as follow :

- We first need to construct the neighbourhood graph. Here, we put an edge between the node i and the node j if these two points are close. To do this, we can choose to put an edge between the two nodes if the distances between the two points is small or if one point is in the set of the k-nearest neighbors of the other item.

- Then, we have to give a weight to the edges on the graph. We can choose simply to give a weight of one to the edge W_{ij} if the nodes i and j are connected or we can give the following weight to the edge that connect two nodes :

$$e^{-\frac{|x_i - x_j|^2}{t}}$$

  where t is a paramater that had to be fixed.

- If we suppose that the graph is connected (if it is not, we have to proceed to the following step for each connected component), we need to resolve the following problem :

$$Lf = \lambda Df$$

  where D is defined by : $D_{ij} = \sum_j W_{ji}$ and $L = D - W$. The matrix L is the Laplacian matrix. This problem has several solutions. As in the principal component analysis, the eigenvectors corresponding to the l biggest eigenvalues give us the new attributes.

2.3.5 Comparisons of the reduction methods

In an article [12], Jarkko Venna and Samuel Kaski realized a comparison of several dimensional reduction methods. First on several typical data sets and then, on what they call an atlas of gene expressions. In this atlas, they had a large collection of human gene expressions. After some transformations and treatments of this data, they obtained a data set containing 1339 genes. That is the data set on which several methods were tested. Some tested methods have not been explained in this review like Isomap or CCA (Curvilinear Component Analysis) which have similarities with MDS. The aim here was not to make an exhaustive list of all the possible methods that can be used to reduce the number of attributes. We will focus on the general results they obtained from the comparison and also to highlight the reason why we will use the principal component analysis to develop the visualization tool for our genes data set.

The comparisons were made using two different measures : one concerning the trustworthiness of the representation and the second one concerning the continuity. These two concepts were defined as follow : a representation is said **trustworthy** if the points that are proximate in the representation are also proximate in the original space. Points that are very close in the original data but that are not very close in the representation decrease the trustworthiness of the representation. Also, a representation is said **continuous** if the points that are proximate in the original space are also close in the representation. If some points are close in the original space

but are not close in the visualization, the continuity of the representation decrease.

Comparison on typical data sets :

The typical data sets on which tests were made have the following shapes :

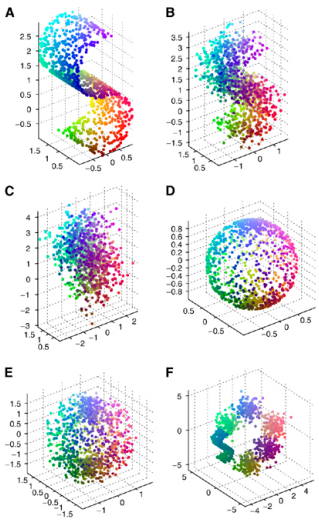


Figure 2.9: Typical data sets tested [12]

Concerning the first comparison on the typical data sets, we will examine their conclusion about the tested methods and studied in this review. For the PCA, they noted that the results in term of continuity were very good and that the results were really easy to understand. But PCA is one of the worst methods in terms of trustworthiness and it can not unfold nonlinear structures. LLE is a good method on simple data sets but that it has big problems to deal with data containing some clusters. And for the last one, the Laplacian eigenmap, they showed that it was one of the best method in terms of trustworthiness. Nevertheless, this method sometimes increase some distances and as a result, the continuity is not very good.

Comparison on genes data sets :

We will now take a closer look to the results they obtained on their atlas of gene expression. They tried, using five methods (PCA, LLE, Laplacian eigenmap, Isomap and CCA), to reduce the number of original attributes (1339) to only two attributes. This task proved to be very difficult since all the data came from different types of experiments. The possibility to find a really good solution with any method was thus very weak. Regarding the trustworthiness of the obtained representation, none of the method gave really good results (which is a proof of the difficulty of the task to reduce the number of attributes from 1339 to 2). Nevertheless, the best results were obtained with the CCA method followed by the Laplacian eigenmap and PCA.

For the continuity measure, the best results were obtained by the PCA followed by the Laplacian eigenmap.

After that, in order to verify their results, they also made the comparison on another gene data set. For the trustworthiness, the best results were obtained by the CCA, Laplacian eigenmap and Isomap. LLE and PCA methods were really bad on this data set. Regarding to the measure of continuity, the best results were obtained by PCA, Isomap and CCA.

Chapter 3

Treatment of the data set

We are now going to explain the solution developed to visualize and explore the gene data set. In this chapter we will focus on the first part of this solution : the treatment applied on the data set for it to be usable with some visualization methods. As we saw, the gene data set on which we are working contains the description of 22.280 attributes that represents gene measurements on 196 patients. This is a too big number of attributes to be usable by some classic visualization methods. This chapter explains how we will reduce this number of attributes. Since the classical dimensional reduction methods are not so good to reduce 22.280 attributes to 2 or 3 (as seen in the section about the comparisons of the dimensional reduction method), we will adapt this dimensional reduction step.

To begin, we create some groups of attributes. Each group contains a set of similar attributes and is represented by an interval attribute that summarize this set of attributes. If we want to have significant groups, we have to keep a number of groups that is still too important to be used directly for the representation. We then apply the principal component analysis on the interval attributes we built. All these different steps will be explained in details and justified in this chapter. We will also examine the results obtained with this method on the gene data set from the Westmead Children's hospital.

3.1 Construction of groups of attributes

When we observe the data set, we can notice that some attributes seem to have similar values. The idea here is to make some groups that will represent a subset of similar genes. Two choices had to be done here :

- **The definition of similar** : which criteria are we going to use to say that two attributes are similar?
- **The use of the groups** : how are we going to take advantage of the definition of groups of attributes to help creating the visualization of the data set?

These two questions are discussed in the following sections.

3.1.1 Basic idea for the construction of the groups

If we want to create some groups of attributes we have to define the criteria that will be used to assign several attributes to a same group.

Before giving the definition of the groups, we need to keep in mind that we develop this method in a very general way. We choose here not to focus on the meaning of the values in the data set to develop a general method. The parameters we will use will thus be very generic. But obviously, these parameters could be adapted if we want the groups to have a more precise meaning or to reflect some particularities of the data set.

Here we want to have, in a same group, attributes that have very similar values or more precisely, attributes that *behave* in a very similar way. To express this, the groups will be defined according to two parameters : the mean and the coefficient of variation.

3.1.2 First parameter : the mean

To express the fact that the attributes are nearly the same, the mean is the first more obvious parameter to use. We define, for each attribute, the mean of the values on each patient as :

$$\mu_j = \frac{\sum_{i=1}^{n} x_{ij}}{n}$$

where μ_j is the mean of the attribute x_j, $j = 1, ..., l$ where l represents the number of attributes and n the number of items (in the case of the gene data set we have $n = 196$).

Once we have the mean of all attributes, we can make a first grouping of the attributes. We will group the attributes that have nearly the same mean. To define this *nearly the same* mean, we will use a certain percentage, p_m, of the scope of the data (that we will note s). The attribute x_j will be in the same group that x_l if :

$$\mu_l - (p_m * s) < \mu_j < \mu_l + (p_m * s)$$

The choice of the scope and the value of p_m will be discussed later.

But if we just use this parameter, we can have, in the same group, attributes that have nearly the same mean but maybe a different behaviour. To highlight this fact, we can examine the two very simple attributes x_1 and x_2 measured on three items :

$$x_{11} = 3, x_{21} = 5, x_{31} = 16$$

$$x_{12} = 7, x_{22} = 9, x_{32} = 8$$

These two attributes have the same mean : $\mu_1 = \mu_2 = 8$. But can we say that these two attributes behave in the same way? The answer is obviously no because the first attribute have a value on the third item that is very different from the values

on the two first items. But for the second attribute, the values on the three items are all around the same value, the value of the mean.

That is the reason why we are going to add a second parameter to the definition of our groups : the coefficient of variation.

3.1.3 Second parameter : the coefficient of variation

The coefficient of variation will allow us to have an idea of the dispersion of the data around the mean. The coefficient of variation of the attribute x_i is defined by :

$$CV_i = \frac{\sigma_i}{\mu_i}$$

where σ_i is the standard deviation of the attribute x_i and μ_i the mean of x_i.

This coefficient is only defined for mean that are not equal to zero which is the case in the gene data set (all the values in the data set are strictly positive).

The choice of this parameter instead of the simple standard deviation is justified by the fact that the standard deviation always have to be analyzed in the context of the mean. The coefficient of variation is a dimensionless number, it does not depend on the mean or on the unit used to measure the values. As the mean of the different genes could be very different, the choice of the coefficient of variation over the standard deviation seemed better.

As for the mean, we will group two attributes if they have nearly the same coefficient of variation but this time, it is not useful to define the *nearly the same* with the scope of the data. Indeed, the coefficient of variation does not depend on the mean and so, on the exact values taken by the data. Here, we will simply allow a certain value of variation of the coefficient of variation, p_{CV}. That is why we will say that the attribute x_j will be in the same group that the attribute x_l if :

$$CV_l - p_{CV} < CV_j < CV_l + p_{CV}$$

With these two parameters used for the construction of the groups we can have exactly what we want, the attributes of a group will behave in the same way : they will have similar means and the dispersion of the values of the attributes around that mean will be similar too.

3.1.4 The scope of the data as a reference for the mean

We saw that in order to express the acceptable percentage of variation for the mean to create the groups, we use the concept of scope of the data. As a recall, the scope is defined as the difference between the maximum and the minimum values we can

find in the data set.

The justification of this choice lies in the fact that, if the data has a high scope, it means that the different attributes may have very different values and thus, very different means. Also, if the data has a small scope, it means that the attributes have very similar values and so, the means may also be very similar.

Moreover the idea to group the attributes is to decrease their number. So, if we want to decrease this number, we need to accept a higher difference for the means if the scope of the data is high. But also, if the scope is small, we need to take a smaller difference to avoid that all the attributes will be assigned to the same group. To take this fact into account, the simplest way was to accept a certain percentage of the scope instead of taking a fixed value as we did for the coefficient of variation.

3.1.5 The choice of the values for p_m and p_{CV}

The choice of the values for p_m and p_{CV} was made by several tests. The idea behind the choice of these values is that we did not want to have too many groups because if the number of groups was still high, the construction of the groups would be less useful. But we did not want either to have a too low number of groups since each group would contain too many attributes, and the meaning of each group would be less interesting.

So, we had to find values that were not too high (to avoid having a small number of big groups) but also not too low (to avoid having too many small groups). We also needed to find an equilibrium between the value of p_m and the value of p_{CV} knowing that :

- to increase (decrease) the value of p_m would have the effect to assign to a group attributes that are less (more) similar in terms of the means.
- to increase (decrease) the value of p_{CV} would have the effect to assign to a group attributes that are more (less) variable in their values.

3.1.6 Optimization : The use of the correlation in a group

Once we built the groups, we made a last optimization before starting using the groups to construct some new attributes. In this optimization we wanted to detect in each group if there was a relation between some attributes in the groups and if we would also try to characterize this relation. To do that, we used the coefficient of correlation.

As a recall, the coefficient of correlation of two attributes (X and Y) give us an information about the relation between these attributes. It can be computed as follow :

$$r(X,Y) = \frac{cov(X,Y)}{\sigma_x * \sigma_y}$$

where σ_x and σ_y denoted the standard deviation of the attributes x and y and $cov(X, Y)$ the covariance between those two attributes.

This coefficient is always smaller than one in absolute value. If r is close from $1(-1)$ it indicates that there is a positive (negative) linear relation between the two attributes. If r is close from 0 then there is no linear relation between the attributes. These cases are summarized on the figure 3.1.

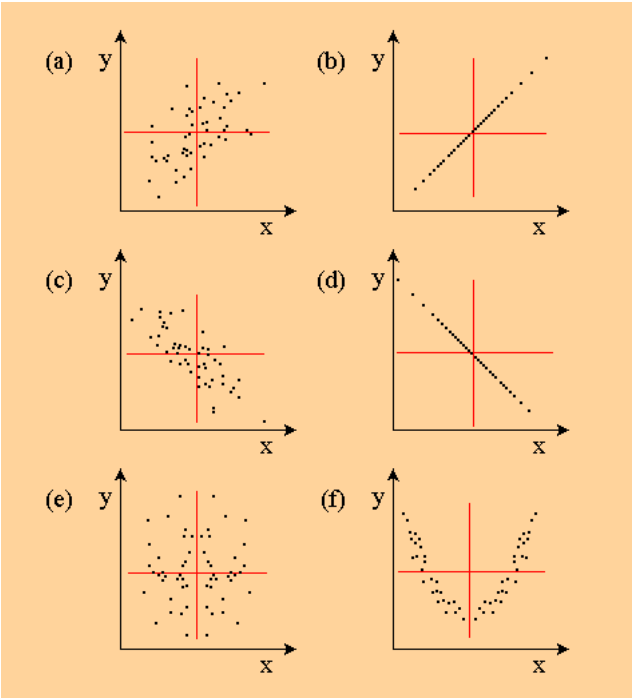


Figure 3.1: Different values of the coefficient of correlation [29]

The plot (a) represents a value of $r = 0.6$, (b) a value of $r = 1$ and we can see the positive linear relation between the two attributes. In the same way, the plots (c) and (d) show values of $r = -0.8$ and $r = -1$ respectively. The two last plots (e) and (f) show that, when $r = 0$, we do not have a linear relation between the attributes on the plot.

What we will try to do in this optimization step is to divide each group into three sub-groups. One that contains attributes that are positively correlated, a second one with the attributes negatively correlated and the last one with the attributes that are no correlated.

Obviously, if the group contains just one attribute, this step will not be done. If the group contains two attributes, we will just check if the two attributes are correlated (positively or negatively) and if not, we will create two groups to separate these

uncorrelated attributes. It is only when we have more than three attributes that we will try to build these sub-groups. From the attributes contained in the groups, we will not always be able to find a sub-group of positively attributes and another one with negatively correlated attributes. Sometimes, maybe we will just have one or two sub-groups.

3.2 Use of the groups : construction of interval data

Now that we have built the groups, we still have to decide how we will use these groups, more precisely, how we will manipulate them. Indeed, for the moment, a group is just a set of attributes with some characteristics. If we want to decrease the number of attributes by using these groups, we have to choose a reference or a summary for these groups. This way, we will be able to consider one group as one new attribute.

3.2.1 General idea behind the construction

Some different options are possible to consider one group as a new attribute. We could decide to take a reference attribute for each group. One group will then be represented by one of its elements. But if we are doing this, the problem will be to choose this element and we are not sure that one element could represent the group correctly as the mean and the coefficient of variation used to build the groups are references. The different attributes assigned to this group represented by just one element as the reference is chosen randomly. We could think to use the element that correspond to the mean and coefficient of variation used as references. If we do that, we will still not have an idea of the other attributes and the way their values are situated around these reference values. The reference mean may corresponds to the attribute with the highest mean in the group and all the other attributes may have a lower mean for example.

Furthermore, if we summarize the groups by just one attribute we are losing the information contained in the other attributes. The final aim of our visualization and exploration tool is to be able to find back to the original attributes. We then need to keep a *summary* of the attributes included in a group, not just one element.

3.2.2 The symbolic data

To make a sort of summary of all the attributes inside a group, the idea is to use the symbolic data and more precisely the interval data. But before explaining how we will build these interval data, we are going to make a short review of this type of data.

Most of the time, we are working on "classical data". These data are defined in an "individuals" by "attributes" matrix where each cell (i, j) contains the value taken by the individual i on the attribute j [21]. This value is said to be *atomic* : it is not a list nor a set of values. But sometimes, such data are not enough to describe a situation.

To illustrate this, we will consider three examples [11] :

- **Symbolic data for individuals** : Suppose we want to describe the daily activities of a consumer k and more precisely the variable Y : *"time spent for watching television per day"*. This attribute can not be described by a single value since the value varies from day to day. To describe this attribute with a single value for each item we can use an interval : $Y(k) = [0, 3]$ in hours, or again a discrete distribution : $Y(k) = ((0, 0.5), (1, 0.4), (2, 0.05), (3, 0.05))$ that can be read as 50% of probability not watching television, 40% to watch the television one hour,...

- **Symbolic data for classes of individuals** : Suppose we do not want to describe each individual but that we want to describe some classes of individuals. Following this idea, suppose we want to describe the attribute Y : *official language* in the European countries. We do not want to have the description of this attribute for every inhabitant of every country but we want to define this attribute for each country. An atomic value is not enough to define the values of this attribute. We may use a set of values $Y(k) =$ {Dutch, French, German} for $k =$Belgium or again, a frequency distribution $Y(k) = ((\text{Dutch}, 0.50), (\text{French}, 0.45), (\text{German}, 0.05)$ which means that 50% of the Belgian speak Dutch,...

- **Vague, uncertain or probabilistic variable values** : We can also use some symbolic data if we want to include some inaccuracy, uncertainty or plausibility in the data. For example, we can define the share price Y of a stock market :
  $Y < 120$ with probability 40%
  $120 \leq Y \leq 130$ with probability 30%
  $130 \leq Y \leq 140$ with probability 20%
  $Y \geq 140$ with probability 10%

These examples allow us to have a general idea of the meaning of the symbolic data. We saw that it is not like in the classical data where the data can be described in a matrix where each cell contains an atomic value. In the case of the symbolic data, each cell will contain a set of values (multi-valued variables), an interval (interval variables) or a probability distribution (modal variables).

We can end this section by giving a last example of each type of symbolic data [21] :

- **Multi-valued variables** : in this case we can have quantitative values, for example $Age =$[15,22,45,47] which means that the age of family members are $15, 22, 45$ and 47 or categorical values as $TVpreference = \{RAI1, R4\}$

- **Interval variables** : $Age =$[15,47] which means that ages in the family are between 15 and 47

- **Modal variables** : $TVpreference = \{(RAI1,0.3),(R4,0.7)\}$ which means that 30% of the family has a preference for RAI1 and 70% for the R4 channel.

In our case, we will use some interval data to describe our groups. One group will be represented by an interval variable. This way, we will have only 231 interval variables on our data set instead of the 22280 original classical attributes. The advantage to use the intervals is that we will be able to keep an idea of all the different values of the original attributes.

3.2.3 The construction of interval data from the groups

To transform each group into an interval attribute, we search, for each patient, in the measures of the genes contained into one group, the minimal and maximal values. These two values will be the bounds of our interval data.

We can illustrate this transformation on the simple following example. Suppose we have six original attributes that have been measured on 4 items. We obtain the following results :

	a_1	a_2	a_3	a_4	a_5	a_6
i_1	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}
i_2	x_{21}	x_{22}	x_{23}	x_{24}	x_{25}	x_{26}
i_3	x_{31}	x_{32}	x_{33}	x_{34}	x_{35}	x_{36}
i_4	x_{41}	x_{42}	x_{43}	x_{44}	x_{45}	x_{46}

If the attributes a_1, a_3 and a_6 have been assigned to the first group G_1, the attributes a_2 and a_4 to the second group G_2 and the attribute a_5 to the group G_3. We will construct three new attributes I_j corresponding to the group G_j.

To compute the bounds of the interval values of the new interval variables, we need to search, in the row of each item and the column of the attributes of a group, the minimum and maximum values. For example, for the first group, we need to search in the following values :

	a_1	a_3	a_6
i_1	x_{11}	x_{13}	x_{16}
i_2	x_{21}	x_{23}	x_{26}
i_3	x_{31}	x_{33}	x_{36}
i_4	x_{41}	x_{43}	x_{46}

The green values are the minimal values and the red ones are the maximal values.

We will then have the following interval variable :

	I_1
i_1	$[x_{11}, x_{16}]$
i_2	$[x_{23}, x_{26}]$
i_3	$[x_{31}, x_{33}]$
i_4	$[x_{46}, x_{41}]$

When a group contains just two attributes, the interval is simpler to build : for an item, the lower (upper) bound is the minimal (maximal) value of the two. And if a group contains just one element, for each item, the lower bound equals the upper bound and the value is the only value we have. We can summarize the results obtains on our six attributes in the following table which summarizes the different cases :

	I_1	I_2	I_3
i_1	$[x_{11}, x_{16}]$	$[x_{14}, x_{12}]$	$[x_{15}, x_{15}]$
i_2	$[x_{23}, x_{26}]$	$[x_{22}, x_{24}]$	$[x_{25}, x_{25}]$
i_3	$[x_{31}, x_{33}]$	$[x_{32}, x_{34}]$	$[x_{35}, x_{35}]$
i_4	$[x_{46}, x_{41}]$	$[x_{42}, x_{44}]$	$[x_{45}, x_{45}]$

When we apply this method on our 231 groups we obtain 231 interval variables. But for the visualization, this number is still too high.

What we are going to do before finally starting with the visualization methods is to apply the principal component analysis. Now that we have 231 attributes instead of 22280, we can hope having better results in terms of variance that the principal component analysis will be able to explain. The choice of this method instead of another one is that, as seen in the comparison of the different methods, the results of the principal component analysis are easily understandable by a user. Since our tool had to be usable by some users not being computer scientists or mathematicians, this method was preferred.

3.3 Principal component analysis for interval data

We want to apply the principal component analysis on our new attributes but since we are now working with interval data, we need to apply the generalization of that method to interval data.

There is two methods to generalize the principal component analysis to interval data : the vertex method and the centers method. They differs in the way data are represented : by their vertex or by their centers. Indeed, interval data can be represented by hyper-rectangles. For example, if we consider an item described by two interval data, we can represent this item by a rectangle. The different possibilities to represent the interval data will be studied in the next chapter. The following figure shows the representation of an item that was describe by two interval data ($I1$ and $I2$) and the values on these two attributes are $[x1, x2]$ and $[y1, y2]$.

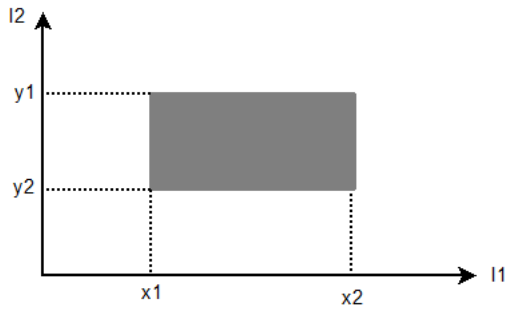


Figure 3.2: An item on which two interval variables are measured

To work on interval attributes and that the items are represented by hyper-rectangles, we have two choices : we can represent this item by the vertex or by the center of the hyper-rectangle. Both methods first consist applying the classical principal component analysis on the vertex or the centers. After that, the interval principal components are built from the results of the classical principal components.

To choose between the two methods, we can notice that, with the vertex methods, we will have to work on a bigger matrix. For each item, we will have to treat each summit. In the case of our interval data, it will means that each item will be represented by 2^{231} summits. We will have to apply the principal component analysis on a matrix of size $n * 2^{q_i} * p$ where n is the number of items (196 in our case), q_i is the number of interval variables describing the item and p is the number of interval variables interfering in the description of the items. In the case of the gene data set, in the worst case, q_i will be equal to 231, as we can have interval variables that are in fact, just a point and p equals 231. Indeed, we will have, in the matrix, all the summits for each item and each summit will have p coordinates.

To illustrate this, we consider 3 items on which three interval variables are measured. The first one has the following values : $([\underline{x}_{11}, \overline{x}_{11}], [\underline{x}_{12}, \overline{x}_{12}], [\underline{x}_{13}, \overline{x}_{13}])$. This item can then be represented by the following matrix that correspond to the summits coordinates :

$$M_1 = \begin{pmatrix} \underline{x}_{11} & \underline{x}_{12} & \underline{x}_{13} \\ \underline{x}_{11} & \underline{x}_{12} & \overline{x}_{13} \\ \underline{x}_{11} & \overline{x}_{12} & \underline{x}_{13} \\ \underline{x}_{11} & \overline{x}_{12} & \overline{x}_{13} \\ \overline{x}_{11} & \underline{x}_{12} & \underline{x}_{13} \\ \overline{x}_{11} & \underline{x}_{12} & \overline{x}_{13} \\ \overline{x}_{11} & \overline{x}_{12} & \underline{x}_{13} \\ \overline{x}_{11} & \overline{x}_{12} & \overline{x}_{13} \end{pmatrix}$$

The second item has the following values : $([\underline{x}_{21}, \overline{x}_{21}], [\underline{x}_{22}, \overline{x}_{22}], [x_{23}, x_{23}])$. Here, the third value is an interval but the lower bound equals the upper bound. The summits matrix is then :

$$M_2 = \begin{pmatrix} \underline{x}_{21} & \underline{x}_{22} & x_{23} \\ \underline{x}_{21} & \overline{x}_{22} & x_{23} \\ \overline{x}_{21} & \underline{x}_{22} & x_{23} \\ \overline{x}_{21} & \overline{x}_{22} & x_{23} \end{pmatrix}$$

The last item has the values : $([x_{31}, x_{31}], [x_{32}, x_{32}], [\underline{x}_{33}, \overline{x}_{33}])$ and then the summits matrix :

$$M_3 = \begin{pmatrix} x_{31} & x_{32} & \underline{x}_{33} \\ x_{31} & x_{32} & \overline{x}_{33} \end{pmatrix}$$

With just three items and three interval variables, we will have to apply the principal component analysis on the matrix M formed with the three summits matrix :

$$M = \begin{pmatrix} M_1 \\ M_2 \\ M_3 \end{pmatrix}$$

If we choose the centers method, we just have to describe an item by the center of the hyper-rectangle that represents it and we will just have to apply the principal component analysis on a matrix of size $n * p$. If we consider our previous example, we will have to apply the principal component analysis on the following matrix :

$$M = \begin{pmatrix} \frac{\underline{x}_{11}+\overline{x}_{11}}{2} & \frac{\underline{x}_{12}+\overline{x}_{12}}{2} & \frac{\underline{x}_{13}+\overline{x}_{13}}{2} \\ \frac{\underline{x}_{21}+\overline{x}_{21}}{2} & \frac{\underline{x}_{22}+\overline{x}_{22}}{2} & x_{23} \\ x_{31} & x_{32} & \frac{\underline{x}_{33}+\overline{x}_{33}}{2} \end{pmatrix}$$

We then choose to use the center method as the number of interval attributes and the number of items will imply a very big size for the matrix in the summits method. The method of the centers can be formalized as follows :

Suppose each item $x_i = (x_{i1}, x_{i2}, ..., x_{ip}) = ([\underline{x_{i1}}, \overline{x_{i1}}], ..., [\underline{x_{ip}}, \overline{x_{ip}}])$ is described by an hyper-rectangle R_i. We have to compute the center $\overline{c_i} = (x_{i1}^c, ..., x_{ip}^c)$ of this hyper-rectangle where :

$$x_{ij}^p = \frac{\underline{x_{ij}} + \overline{x_{ij}}}{2}$$

We need to compute the center of each item to have a centers matrix :

$$X = \begin{pmatrix} x_{11}^c & \cdots & x_{1p}^c \\ \vdots & \vdots & \vdots \\ x_{n1}^c & \cdots & x_{np}^c \end{pmatrix}$$

We then need to apply the principal component analysis on the centers of the items which means we apply the principal component analysis on the matrix X.

After that, we can compute the values of the l-th interval principal components on the item i by computing :

$$\underline{y_{il}} = \sum_{\{j|v_{jl}<0\}} \overline{x_{ij}} - \overline{x_j^c} + \sum_{\{j|v_{jl}>0\}} \underline{x_{ij}} - \overline{x_j^c}$$

$$\overline{y_{il}} = \sum_{\{j|v_{jl}<0\}} \underline{x_{ij}} - \overline{x_j^c} + \sum_{\{j|v_{jl}>0\}} \overline{x_{ij}} - \overline{x_j^c}$$

where v_l is the l-th eigenvector of the covariance matrix of the centers.

3.4 The general algorithm

3.4.1 First construction of the group

To implement the construction of the groups, we still need to make some choices. The first one concerns the mean and the coefficient of variation that we will take as reference for the construction of the groups. Indeed we want to group the attributes with these two parameters but following the reference values taken for each group, the grouping could be very different. There is a lot of different solutions that can be developed in this part of the algorithm.

The solution that we will keep for the implementation is to read the attributes in the order they appear in the data file. Each time we read one attribute, we compute its means and coefficient of variation. We then try to assign it to an existing group. If it is not possible to put it into an existing group, we create a new group and the mean as well as the coefficient of variation of this attribute are used as references for this group. The other attributes will be compared to these values of mean and coefficient of variation. Like this, the first attribute we read can not be assign to a group since there is still any group and thus the first group is created with the mean and the coefficient of variation of this first attribute. After this, we will read each attribute in the file and assign it to an existing group or to a new group created for it.

This approach is very simple and we could have taken a different one. Indeed, we could have chosen to take a random attribute to read each time in the file instead of just reading the file in order. This approach would have the advantage to avoid a possible existing order in the data file. Indeed, in the data file, we could have a certain order in the attributes having an influence on the groups that we will realize. If an attribute with a certain mean and a great coefficient of variation is in the beginning of the file, this coefficient of variation will be taken as a reference. Maybe more attributes will be assigned to that group because of this great value. Depending on the information represented in the data, we could prefer this approach to break the order in the presentation of the data. But here, as we did not take into account the meaning of the data and the construction method of the data file, the simple approach was chosen.

We could also run the construction of the groups several times until we obtain the best grouping. This approach could be better if we wanted to optimize a certain criteria. But also, if we did not want to take care about the meaning of the data, we could not define a criteria to optimize. We can just try to define an expected structure in the groups or define an expected number of groups.

```
//loop on the attributes of the data set
for i = 1, ..., n
 read attribute x_i
 compute mean μ_i
 compute coefficient of variation CV_i
 assign = false

 //loop on the existing groups
 for j = 1,...,l
     if μ_j − p_m < μ_i < μ_j + p_m
       then if CV_j − p_CV < CV_i < CV_j + p_CV
            then assign x_i to G_j
                 assign = true
                 break
 end for j

 if assign = false
      create a new group for  x_i  with  μ_i, CV_i  as references
end for i
```

At this stage, we have the first version of the groups. The optimization is the next step to be done.

3.4.2 Optimization of the groups

As we saw in a previous section, we need here to check every group we previously created. For each group, we will try to divide them into three sub-groups (if the group contains more than three attributes). As we are working with the coefficient of correlation, we begin by computing the correlation matrix on which we will be working.

Before beginning the explanation of the implementation of this part, we have to define the values of the coefficient of correlation from which the attributes will be considered as correlated (positively or negatively). The threshold values that we will take will be 0.75 and -0.75. More exactly, we will state that two attributes are positively correlated if their coefficient of correlation is greater than 0.75. Two attributes are negatively correlated if their coefficient of correlation is less than -0.75. Otherwise, these two attributes are considered as non correlated.

The general aim would be to find a sub-matrix in the correlation matrix that contains just values greater than 0.75 or just values smaller than -0.75, which would indicate that the attributes are positively or negatively correlated. Since this problem would take too much time to compute, we will do this another way. To begin, we will compute, for each row, the number of coefficient of correlation that are positive

and we do the same for the negative coefficient.

```
loop on the row of the matrix of correlation C
for i = 1,...,n
     //We define two tables
     //One that contains the number of maximum correlation
     maxCorr = new int[n]
     //And another one with the number of minimum correlation
     minCorr = new int[n]
     for int j = 1,...,n
          if C(i,j)>0.75 then maxCorr[i] = maxCorr[i]+1
          if C(i,j)<-0.75 then minCorr[i] = minCorr[i]+1
     end for j
end for i
```

After this, we put in the group that will contain the positively correlated attributes, the attribute corresponding to the biggest number of positive correlation with other attributes. We also do the same for the negative correlation. This way, we have the first reference elements for our two temporary sub-groups. Afterwards, we will take the attributes that corresponds to the biggest number of positive or negative correlation in decreasing order. We will check if their correlation with the elements already in the groups of maximum (minimum) correlation is bigger than 0.75 (smaller than -0.75). The reason to consider the attributes in decreasing order for the number of attributes they are correlated with is to try to have a maximum chance for that attributes to be correlated with some other attributes. Indeed, if we take a second attribute in one of the subgroups and that this attribute is only correlated with the first attribute, we will not be able to add any other attribute in this subgroup.

```
//We need a set with the elements positively correlated
tempGroupMaxCorr = {}
While  ∃i as maxCorr[i] ≠ 0
     find j as maxCorr[i]> maxCorr[j] j =, ..., n j ≠ i
  for j ∈ tempGroupMaxCorr
         if C(i,j)>0.75 add i to tempGroupMaxCorr
  end for j
end while

//We need a set with the elements negatively correlated
tempGroupMinCorr = {}
While  ∃i as minCorr[i] ≠ 0
     find j as minCorr[i]> minCorr[j] j =, ..., n j ≠ i
  for j ∈ tempGroupMinCorr
         if C(i,j)<-0.75 add i to tempGroupMinCorr
  end for j
end while
```

After this we have to check that some attributes are not in both sub-groups of positive and negative correlation. If it is the case, we choose to leave the attribute on the positive correlation group. This is a deliberate choice to do so. We could have decide to leave them in the negative correlation sub-groups, to leave them in one of the sub groups alternatively or again to randomly leave them in one of the two groups.

The attributes who does not appear in one of the two sub-groups are assigned to the non-correlation sub-group. For the moment, we just worked on the index of the attributes. We then need to construct the structures of the groups.

A last thing to do is also to examine if some groups created this way contain more than just one attribute. Indeed, as we are assigning one element in the sub-sets with positive and negative correlation, if any argument is added to these groups, it is not useful to keep them. The attribute in this sub-group is added on the no-correlation group.

The algorithm can be described as follow :

```
//We are working on the groups previously build :
G = {G_1, ..., G_l}
//We need a set that will contains the final groups :
GFinal = {}
//Loop on the groups from G
for int i = 1, ..., l

     if size of G_i = 1
          add G_i to G
     end if

     if size of G_i = 2
          G_i = {x_1, x_2}
          compute r(x_1, x_2)
          if r(x_1, x_2) > 0.75 or r(x_1, x_2) < -0.75
             add G_i to G
          else create two groups G_i^1 with x_1 and G_i^2 with x_2
     end if

     if size of G_i > 2
          size of G_i = m
          tempGroupMaxCorr = {}
          tempGroupMinCorr = {}
          tempGroupNoCorr = {}
          compute the correlation matrix C
          compute maxCorr[m]
          compute minCorr[m]
          while ∃i as maxCorr[i] ≠ 0
               search biggest element in maxCorr  x_M
               if r(x_M, x_l) > 0.75 ∀x_l ∈ tempGroupMaxCorr
                    add this element to tempGroupMaxCorr
               end if
          end while

          while ∃i as minCorr[i] ≠ 0
               search biggest element in minCorr  x_m
               if r(x_m, x_l) < -0.75 ∀x_l ∈ tempGroupMinCorr
                    add this element to tempGroupMinCorr
               end if
          end while
          if common element between tempGroupMaxCorr and tempGroupMinCorr
               let this element in tempGroupMaxCorr
          end if
```

```
          if size of tempGroupMaxCorr == 1
               add attribute of tempGroupMaxCorr to tempGroupNoCorr
          end if
          if size of tempGroupMinCorr == 1
               add attribute of tempGroupMinCorr to tempGroupNoCorr
          end if
     end if
end for
Construct the groups structure for each groups in GFinal
```

3.4.3 Construction of the interval data

Now that we have our groups, we can build our interval data. In order to do so, we saw that we needed to find the lower and upper bound of the interval which represent the lower and upper values of all the attributes contained in one group for all the items.

```
GFinal = {G_1, ..., G_l}
for i = 1,...,l
     Extract all the attributes x_1, ..., x_n of the groups
     //Each attribute is measured on k items x_j = [x_j1, ..., x_jk]
     //We need to compute the value of the interval attribute I_i = [I_i^1, ..., I_i^k]
     for s = 1,...,k
          find min values of x_js for j=1,...,n x_Ms
          find max values of x_js for j=1,...,n x_ms
          I_i^s = [x_ms, x_Ms]
     end for
end for i
```

3.5 Results for the gene data set

Now that we saw the detailed explanation of how we reduced the number of attributes of the gene data set, we will examine the results obtained and also examine the different choices we made for some values.

3.5.1 The parameters for the mean and the coefficient of variation

The first thing to do is to build the first version of the groups. In order to do that, we first need to fix the percentage of variation for the mean and the coefficient of variation we will use. We saw that we can fix these values by testing different possibilities in order to find an equilibrium between these two values to reach a certain number of groups.

The reference approximate number of groups used as a reference in the tests was 200. After some tests, the following values were chosen :

$$p_m = 0.15$$

$$p_{CV} = 1$$

knowing that the scope of our gene data set is 31702.16 which means that the means are considered to be nearly the same if the difference between them is smaller than 48.

3.5.2 Number of groups before the optimization

With these values, we built 184 groups. In this 184 groups, 58 contained only one attribute. These groups were very interesting because if an attribute was the only one to be assigned to a group, it means that this attribute was more different from the other ones. Indeed, any other attribute had a similar mean or coefficient of variation which can indicate that we could focus more precisely on this attribute to discover the meaning of this difference. It can indicate, depending on the meaning of the data, that this attribute is a sort of *outlier* or even that the measures for this attribute were not correctly taken for example (maybe the method used to measure this attribute is not really appropriate).

Another interesting group contains 10493 attributes which means that it contains nearly the half of the attributes (22280 at the beginning). This indicates that more than the half of the attributes behaves in the same way.

We also have a group with 2945 attributes and another one with 2647 attributes. Two other groups contains 1678 and 1171 attributes. Seven groups contains between 100 and 1000 attributes. 29 groups contains between 10 and 100 attributes. The other groups (85 groups) contains less than 10 attributes.

The interesting thing to notice here is that we were able to find in the data some groups that contains a lot of attributes. This can indicate that among the different

attributes, there are some important values around which the attributes are. We also have a lot of groups that contains not so many attributes and maybe these attributes indicate some particularities in the data set.

We can represent the different groups on a scatter plot. The x-axis represents the mean use for the group and the y-axis is the coefficient of variation.

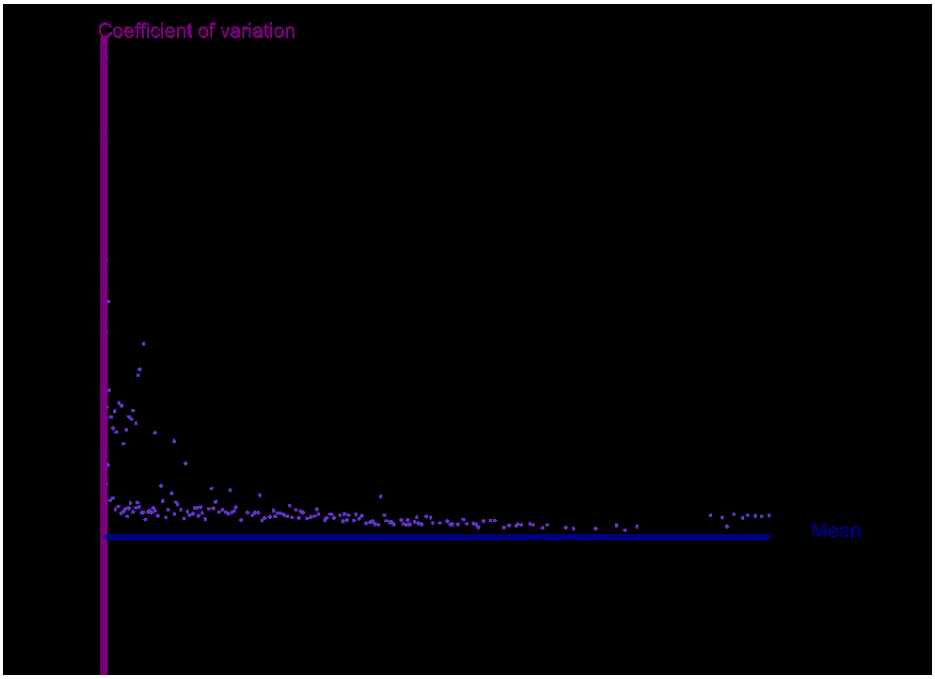


Figure 3.3: Representation of the groups

3.5.3 Number of groups after the optimization

After the first construction of the groups, we try to optimize them by taking into account the correlation inside each group. From the 184 groups during the initial phase of the construction, we build 232 groups which means that 48 new groups have been created.

To begin, we can notice that the number of groups containing only one element did not change much (we now have 61 groups with only one element). This shows that, from the numerous groups containing only two attributes, most of them were correlated. Indeed, the groups containing two elements are the only one that can create two new groups with one element in the optimization.

We can also notice that, from the biggest group, containing 10493 attributes, we were only able to create two subgroups. One big subgroup containing obviously the

non-correlated attributes but also a subgroup that contains 575 attributes that are all positively correlated. This observation tells us that these attributes seem to have a strong positive influence one on the others. But also, the fact that we could not build a subgroup with negative correlation indicates also that any attribute has a negative influence on some other attributes. Or the elements that have a negative influence on some attributes also have a positive influence on some other attributes.

If we look closer to the groups containing a lot of attributes, we can notice that it is almost always the same scenario : from these groups, we nearly create two subgroups : one with positively correlated attributes and another one with groups that are not correlated.

In fact, when we take a closer look to all the groups created during this optimization phase, we can notice that only 2 groups have been created that indicate a negative correlation and these 2 groups are very small since they both contain only two attributes.

3.5.4 Construction of the symbolic data

The next step is to create the symbolic data from these groups. We will then have 232 interval variables each one measured on the 196 items that the gene data set contains. But since there are 61 groups that contain one attribute, we will then have 61 interval variables that are in fact, classical attributes. These particular interval attributes will have the following form :

$$I_i = ([x_{i1}, x_{i1}], [x_{i2}, x_{i2}], ..., [x_{i196}, x_{i196}])$$

3.5.5 Principal component analysis results

Now that we have our interval variables, we can apply the principal component analysis on these attributes. To apply the principal component analysis we have to choose the percentage of variation from the original data that the principal component will explain.

To be able to choose this value, we also made some tests knowing that, if this percentage is too high, the number of principal components would be too high as well. If this percentage was low, we would have less principal components but the meaning of these principal components would not be very interesting. Since the final aim of this dimensional reduction method is to get a number of attributes that will be usable by some visualization method, we fixed the maximum number of principal component to 10. With this limit, we were able to explain 80% of the original variation of the data with only 9 interval principal components.

With this method, we were able to decrease the number of attributes from 22280 original attributes to only 9 new attributes.

We have now 9 attributes and we will be able to use these attributes with some visualization methods. This will be the subject of the next chapter.

Chapter 4

Visualization

Once we have our treated data and a usable number of attributes, we are going to see the various visualization methods that will be used in our tool. One is a simple generalization from the classical scatter plot, the other one had been developed to represent the distances between the items for all the original attributes.

The generalization of the scatter plot that we will call *Multiple scatter plot* will be the starting point of the data exploration. It is from this multiple scatter plot, but also some other visualization methods, that we will choose the part of the data set we want to explore. This part of the tool will be described in the next chapter.

This chapter will also describe the only visualization implemented in the tool that focus on the items rather than on the attributes. Indeed, the choice was made to explore the data set on the attributes side, but we tried to get an idea of the differences that exist between two attributes.

These two visualizations will be presented and discussed in the following section.

4.1 Multiple scatter plot

4.1.1 General idea

The multiple scatter plot is a simple generalization of the classical scatter plot. However, instead of representing 2 or 3 attributes on a 2 or 3D scatter plot, we will represent all our attributes (nine principal components in the case of the gene data set) on a set of 2D scatterplot. Each scatter plot will be placed one behind another with a little shift on each axis to give an insight of each scatter plot.

The multiple scatter plot is thus a 3D visualization with the readability of the 2D scatter plot.

For that representation to be useful, we first need to associate the possibility to browse the graphics and also to move the position of the multiple scatter plot.

Indeed, if we are only able to see all the scatter plots one behind the other, we are limited to get a general insight of the points position on the different scatter plot. It remains difficult to see the exact position of the different points on the various scatter plots forming the multiple scatter plot. We also added some other functions to modify the multiple scatter plot to allow each user to place each attribute at the place he prefers and that is the more significant for him.

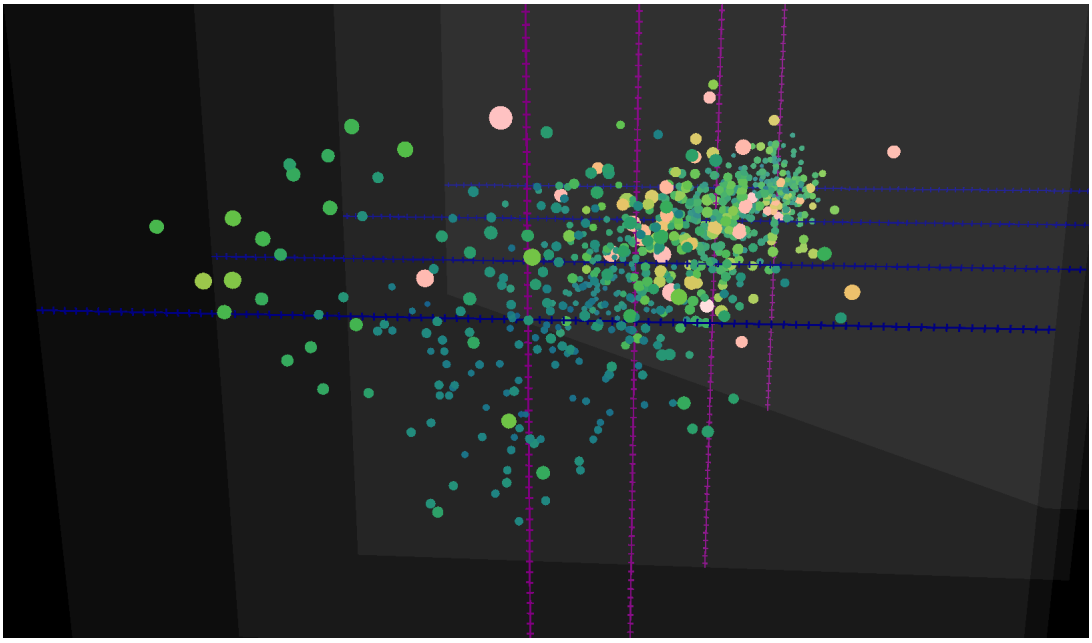


Figure 4.1: Multiple scatter plot

4.1.2 Representation of symbolic data

Before we explain how we will represent an object on the multiple scatter plot we need to make a short review of the different ways we can represent the interval data. This is indeed the type of data that will be shown on the multiple scatter plot. We will consider three different representations : by hyperrectangles, by zoom stars or eventually by Kohonen maps.

Representation by hyperrectangles [11]

Suppose we have an item i described by p interval attributes :

$$x_i = ([\underline{x}_{i1}, \overline{x}_{i1}], ..., [\underline{x}_{ip}, \overline{x}_{ip}])$$

This item can be visualized by an hyperrectangle R_i with 2^p vertices in the space R^p. The lengths of the hyperrectangle sides are given by the span of the intervals

of the corresponding descriptive features. When $p = 2$, this corresponds to a simple rectangle, as illustrated on the following figure :

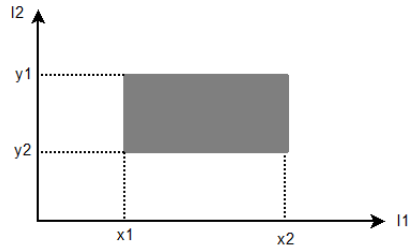


Figure 4.2: Representation of an item described by two interval variables

An hyperrectangle in R^p can be described by a matrix that contains the coordinates of the vertices. This matrix will then have 2^p rows and p columns. For example, the item i in the case where $p = 2$ can be described by the following matrix :

$$M_i = \begin{pmatrix} \underline{x}_{i1} & \underline{x}_{i2} \\ \underline{x}_{i1} & \overline{x}_{i2} \\ \overline{x}_{i1} & \underline{x}_{i2} \\ \overline{x}_{i1} & \overline{x}_{i2} \end{pmatrix}.$$

Representation by zoom stars [11], [20]

The zoom star is a representation that allows us to visualize not only interval attributes, but all the different types of symbolic data. This representation is available in the SODAS software. It is a radial graph that will represent an item described by a set of symbolic attributes. Each symbolic attribute is represented by an axis on the graph. There exists different versions of zoom stars : the 2D zoom star, the 3D zoom star and the temporal star. We are going to examine these different versions.

The 2D zoom star :

This representation is realized in the plane. Each axis is linked according to the value of each variable. If an axis represents a categorical attribute (modal or multi-valuated), all the possible categories is represented on the axis. A dot is placed on each category present in the description of the object. The diameter of the dots represents the frequency of the category. If the attribute is an interval attribute, we represent the limits of the intervals.

All values of the attribute are linked. When we arrived at a categorical attribute, we made the link with the category that had the highest frequency if it is a modal attribute or with each present category if this is a multi valuated attribute. If it is an interval attribute, we make the link with the two bounds of the intervals and the

interval surface is coloured.

If we have a modal attribute, we only see the category with the highest frequency. But if we want to see all the possible categories, we can display the histogram of this attribute.

An example of zoom star is displayed on the following figure :

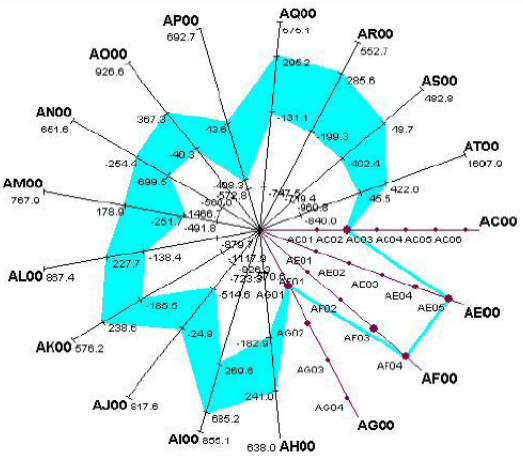


Figure 4.3: An example of 2D zoom star [25]

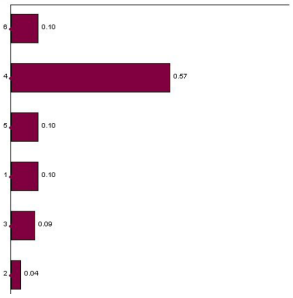


Figure 4.4: Histogram of a modal attribute [25]

Each item can be visualized by a zoom star. If we want to compare different items, we can visualize the different zoom stars side by side, or on the same graph, using superimposition, different colours and transparencies as illustrated on the following figure :

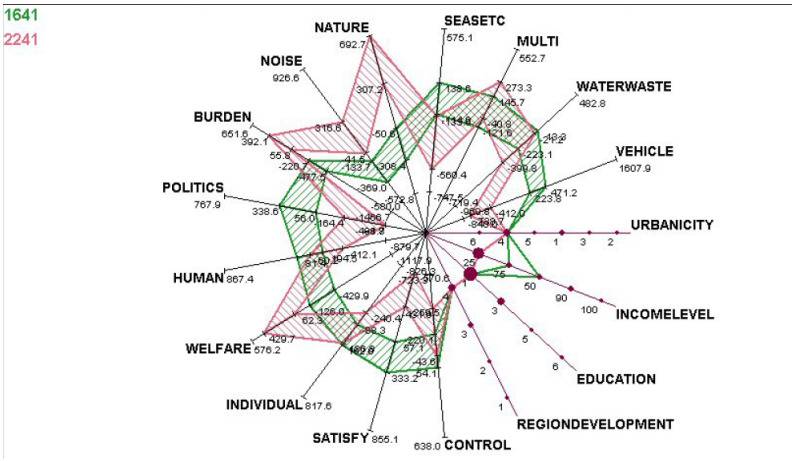


Figure 4.5: Comparisons of two items by superimposition [25]

The 3D zoom star :

The 3D zoom star is a generalization of the 2D zoom star representing the zoom star in 3D rather than in the plane. The third dimension is used to represent the histograms associated to the modal attributes.

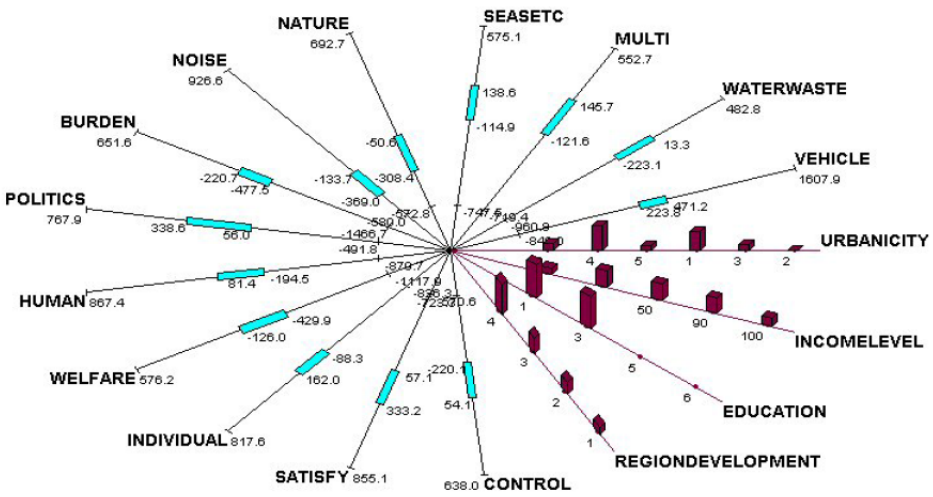


Figure 4.6: An example of 3D zoom star [25]

The temporal star :

The temporal stars are used to show data varying with time. We visualize an item on different times on different 3D zoom stars. These different zoom stars are

represented on a central axis.

Representation by Kohonen maps [9]

We can also represent the items described by interval data with the help of the Kohonen maps. In this representation, instead of showing each item in a smaller dimension space, as we usually do with a dimensional reduction method, we are going to create some "mini-clusters" of items. These clusters will be assigned to the vertices of a fixed, prespecified rectangular lattice L of points in the plane in a way that similar clusters will be assign to neighourood vertices of L.

The data will then be represented by vertices $P_1, ...P_m$ of a rectangular lattice L with b rows and a columns, such that each vertex P_i represents a homogenous cluster C_i of objects and a prototype z_i describing the overall properties of this cluster. During the construction of the clusters and of the prototypes, the aim is to try to represent the neighbourhood structure present in the general representation of the points in R^p in L.

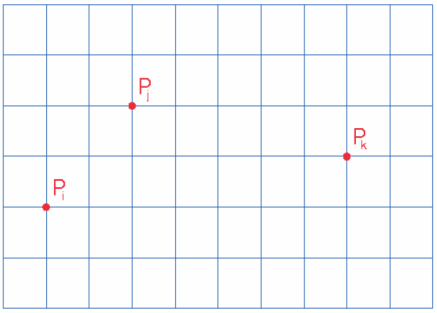


Figure 4.7: An example of lattice [9]

This approach was developed for classical data but a generalization for interval data is present in the SODAS software by the module SYKSOM.

We will not detail each step of the algorithm implemented in SYKSOM since, in our tool, we will not try to create some clusters in the items. The aim of this section is mostly to get an idea of the possible ways to represent the symbolic data.

The construction of the representation can be described with the following steps :

- Each item is represented by an hyperrectangle, as seen in a previous section. Each hyperrectangle is clustered into one of the $m = b * a$ non-overlapping clusters $C_1, ..., C_m$.
- Each cluster C_i is represented by a prototype hyperrectangle z_i in R^p.

- Each cluster and thus each prototype z_i is assigned to one of the vertices of L.
- This assignation is made in a way that if some clusters are neighbour in R^p, they will also be neighbour in L.

The SYKSOM algorithm propose three different methods to build the clusters and the prototypes but these will not be detailed here.

What we still need to know is how the different clusters and thus the different prototypes will be visualized. There are different approaches which correspond to three different modules in the SODAS software :

- VMAP : display the lattice. Each cell, corresponding to one of the cluster contains two icons : a circle that represents the size of the cluster and a square that represents the volume of the corresponding prototype hyperrectangle. This module also includes some options to visualize the clusters by zoom stars or bar diagram for example.

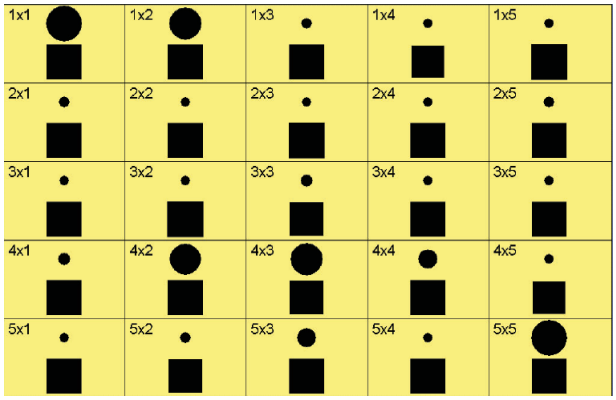


Figure 4.8: Representation of the clusters by a circle and a square [25]

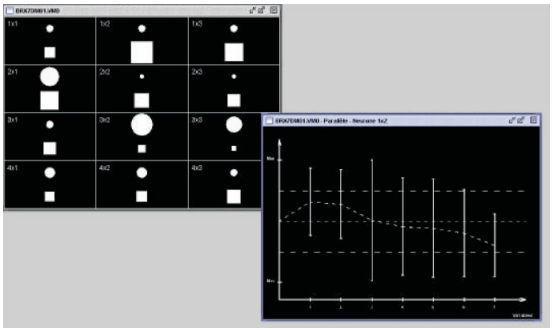


Figure 4.9: The same representation with a bar diagram [25]

- VIEW : This module allows to get the description present in the VMAP module for all the clusters simultaneously.

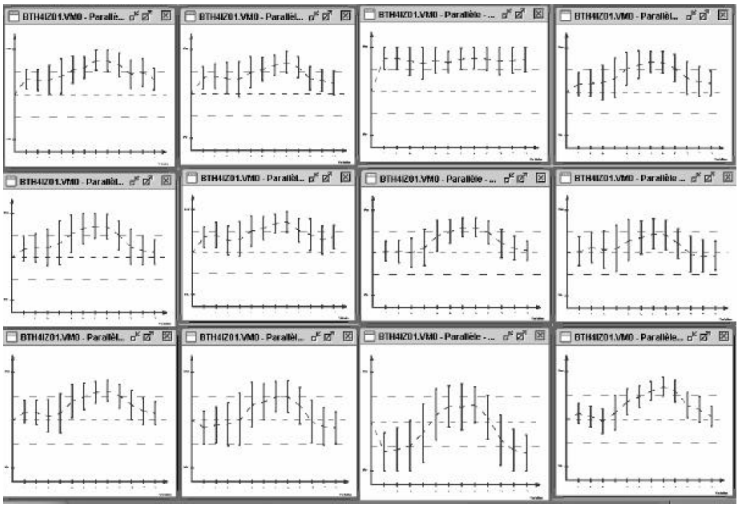


Figure 4.10: Representation of each clusters by a zoom star [25]

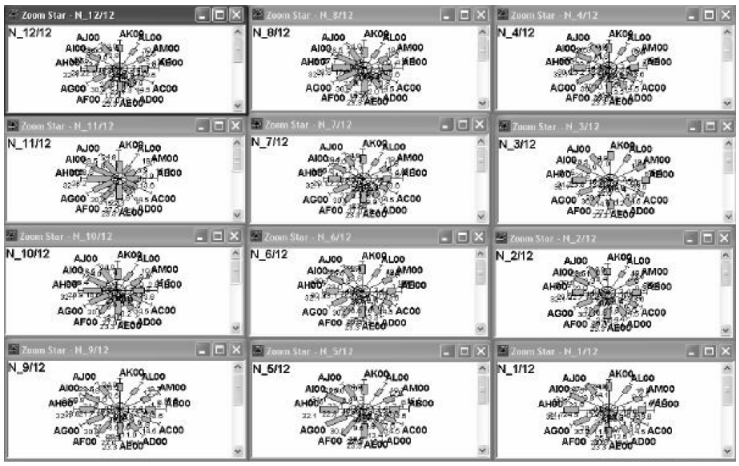


Figure 4.11: Representation of each clusters by a bar diagram [25]

- VPLOT : This module provides a geometrical display of the mini-clusters in the space of two arbitrarily selected interval variables. It corresponds to the projection of the prototypes in the space spanned by the two selected items.

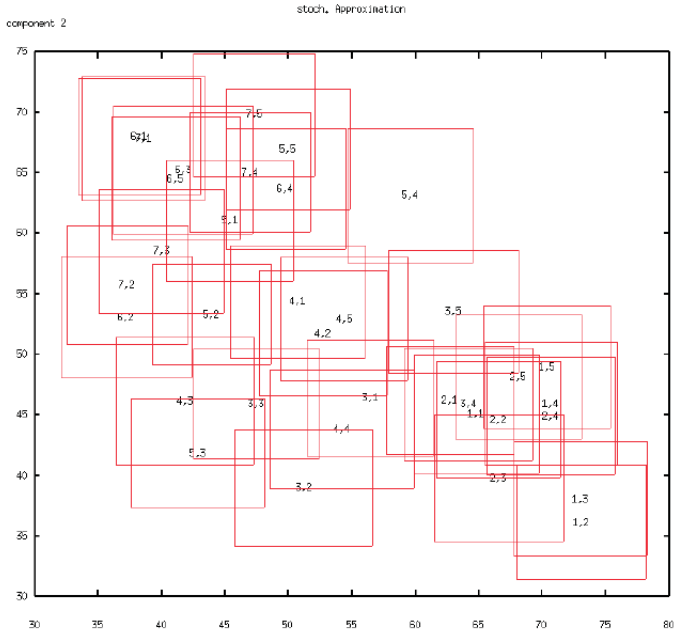


Figure 4.12: VPLOT representation [25]

4.1.3 The representation of the interval value on the scatter plots

We have to keep in mind that the values we want to show on the different scatter plots are interval variables and not only classical variables like we usually represent on a scatter plot. We thus need to adapt the representation of an item on a scatter plot since the representation by a simple point is not enough here. Indeed, we are working with a 2D scatter plot, the items we will show on the scatter plot will then have coordinates of type :

$$x_i = ([\underline{x}_{i1}, \overline{x}_{i1}], \underline{x}_{i2}, \overline{x}_{i2}])$$

If we want to represent such an item on the scatter plot we then need to find a way to represent the position of the rectangle that represents this value and also the span of the two intervals. At one stage of the exploration of the data, we will find back the original attributes that are not interval attributes. The representation we choose for the items had to allow us to represent such classical items afterwards.

In the previous section, we saw some possibilities to represent the interval data. The zoom star is not usable here if we want to show the attributes on a set of classical scatter plots. Moreover, if we choose to represent the items with some zoom stars, we would have had one zoom star by item (which means 196 zoom stars for the gene data set) and this number is too big, even if we use the superimposition (196 items to superimpose is too much).

The representation by the Kohonen maps is also not optimal in this case since we do not want to create some clusters in the items.

We can consider the representation by hyperrecangles but the problem with this representation is that we would have 196 rectangles on each scatter plot and it would be difficult to focus on a particular item if there are so many rectangles. We will then adapt this representation by hyperrectangle to increase the readability of the results.

The idea that we are going to apply here is to use a sphere in a similar way as in the usual scatter plot since each item is represented by a point. But here, we will also use the color and the size of the point to represent the different values of the intervals.

Each sphere will be positioned at the coordinates of his center. If we consider the general item defined sooner, the center of the sphere will be given by :

$$c_i = (\frac{\underline{x}_{i1} + \overline{x}_{i1}}{2}, \frac{\underline{x}_{i2} + \overline{x}_{i2}}{2})$$

Once the sphere is positioned on the scatter plot we still need to represent the spans of the two intervals. In order to do that, we will use the color and the size of the sphere.

The span of the first interval ($[\underline{x}_{i1}, \overline{x}_{i1}]$), corresponding to the x-axis of the scatter plot will be represented by the radius of the sphere. The radius of the sphere will then have the following value :

$$r_i = (\overline{x}_{i1} - \underline{x}_{i1}) * f_d$$

where f_d is a division factor that we use to normalize the size of the sphere. Indeed, the length of an interval can be very big and if we want to avoid having a huge sphere that will cover some other points, we need to fix a maximum size for the sphere and then scale all the interval spans to this maximum size.

In a similar way, to represent the span of the second interval ($\underline{x}_{i2}, \overline{x}_{i2}$) corresponding to the y-axis of the scatter plot, we will use a scale of color. We have a scale of colors and following the span of this interval, we will assign it to a particular color.

Example :

To illustrate this, consider the four following items :

	x_1	x_2
Item 1	$[10, 20]$	$[2, 4]$
Item 2	$[5, 20]$	$[1, 7]$
Item 3	$[3, 18]$	$[2, 5]$
Item 4	$[2, 12]$	$[4, 5]$

As we saw, we will position the sphere at the center of the rectangles describing the interval values. We will then position the item 1 at $(15, 3)$, the item 2 at $(12.5, 4)$, the item 3 at $(10.5, 3.5)$ and the item 4 at $(7, 4.5)$.

Then we will have to describe the span of the intervals of x_1 with the size of the spheres. We can see that the spans of these intervals is 10 or 15. We will choose a division factor of 0.1 to obtain a radius of 1 or 1.5 for the spheres.

For the length of x_2, we will represent it with a scale of color. We will choose to represent the length smaller or equal to 2 with a grey color. The values greater than 2 but smaller than 5 will be represented by a light blue color and the values bigger than 5 will be represented by a dark blue color. We will then have the following values for the items :

	Center	Radius	Color
Item 1	$(15, 3)$	1	Grey
Item 2	$(12.5, 4)$	1.5	Dark blue
Item 3	$(10.5, 3.5)$	1.5	Light blue
Item 4	$(7, 4.5)$	1	Grey

And we get the following representation :

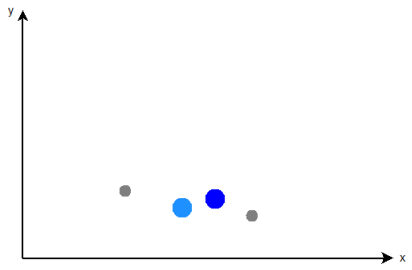


Figure 4.13: Representation of the four items with the size and the color of the spheres

4.1.4 Application to the principal components built on the gene data set

We can apply this visualization to the 9 principal components that we built on our gene data set. But with the 2D scatter plots we can only directly show an even number of attributes : we can set two principal components on each scatter plot. To represent the ninth principal component, we choose to use the first principal component a second time. But as we will see in the next section, after this basic version of the multiple scatter plot, we will be able to modify the position of each attribute. This way we will have 5 scatter plots :

	x-Axis	y-Axis
Scatter plot 1	PC1	PC2
Scatter plot 2	PC3	PC4
Scatter plot 3	PC5	PC6
Scatter plot 4	PC7	PC8
Scatter plot 5	PC9	PC1

4.1.5 Applicable modifications on the multiple scatter plot

To help the user to get more information or to get another view of the different scatter plots, we associate different functions to the multiple scatter plot. This way, we can :

- **change the attributes on each scatter plot** : we can change the association of attributes on each scatter plot. We can, for example, decide that the first scatter plot will be the association of the first and the third principal components.
- **add or remove a scatter plot** : we can also remove one of the scatter plot or add a new one.
- **select a scatter plot** : if we want to have a better view of one of the scatter plot, we can select it.
- **see an item on every scatter plot** : if we want to focus on a particular item, we can select this item and see just this item on the multiple scatter plot.

This way, we have an adaptative representation that a user can configure the way he wants.

4.1.6 Discussion

With this visualization, we can have a simple view of the data. The 2D scatter plots are easily readable and with the use of the colors and the size of the sphere it stays

readable, even with interval attributes. If we add the possibility to move inside the visualization and also other functions as we described, we can also easily see each different scatter plots.

We can see that, by using a scale of colors and the size of the sphere, we do not have directly the exact value of the two intervals. But to fix this problem, we have the possibility to display the exact values of all the principal components.

With the multiple scatter plot, we can represent only a small number of attributes. Indeed, if we want to keep the global representation readable, we can not display as many attributes as we want. In the tool, the number of scatter plots on the multiple scatter plot is limited to 5. But if a user wants, he can add as many scatter plots as he wants. This limitation is not a real problem in our case since we try to limit the number of attributes that a user will see in one time.

Maybe the use of a 3D visualization may seem more difficult than a 2D visualization. But we have the possibility to choose one scatter plot at a time. This way, we can just have a 2D scatter plot.

4.2 Level circles representation

4.2.1 General idea

The idea with this representation is to see the main differences existing between two items. Indeed, we saw that the genes data set contains measures of genes that were taken on sick but also healthy patients. Maybe it would be interesting to see which genes are very different between a patient who is sick and a healthy one. More generally, we want to find a way to see whether there is a big distance or not between the values of a gene taken on two different patients. Maybe this could highlight some particular genes that are more important to explain a disease or some other things.

We will not represent here the attributes that we built in the dimensional reduction phase but the original attributes. We thus need to find a simple and readable way to represent the distances for each original attributes which means 22280 attributes.

The approach taken here is to work in levels which will be displayed one above the other. On each level we will represent a certain number of attributes and these attributes will be organized in a circular way. Each attribute will be represented by a cylinder on a level. To form the *circle*, we will draw a first cylinder (a first attribute). To draw the second one, we will put the second cylinder in the same plane than the first one but we will add a rotation around the y-axis to that second attribute. We will then do the same for each attribute of that level. The angle of the rotation will be increased each time.

Formally, suppose we want to have n attribute on a level, the first attribute will not be rotated but the second one will be rotated by an angle of :

$$\frac{2\pi}{n}$$

Indeed, if we want to display n attributes, we need that the n cylinders form a circle on a level, each cylinder had to be rotated by this angle. This way, the $i-th$ attribute we want to display will be rotated by an angle of :

$$(i-1) * \frac{2\pi}{n}$$

And the last one :

$$(n-1) * \frac{2\pi}{n}$$

the last attribute we display on the level will be the neighbour of the first one, which corresponds to a rotation of 2π (or no rotation).

4.2.2 The choice of the represented distance and normalization of these distances

Now that we explained the way we will organize the cylinders on a level, we just need to see how we will represent the distances between the values for two items of an attribute. This will simply be done by taking the difference between these two values. Like we did with the radius of the sphere, we will also normalize these values to scale them in the interval $[0, 5]$.

These values have been chosen after several tests to have a good visualization even when we have more levels. Indeed, for an attribute being visible, we need it to have a sufficient length in comparison of the height of the representation.

We choose here as the distance the simple difference between the two attributes but if we have to take into account the meaning of the data, maybe it would have been more useful to work with another distance.

4.2.3 The choice of the number of attributes by level

We need to choose the number of attributes that we will represent on a level. This number has to be quite big, to be able to represent a sufficient number of attributes on a level but it should not be too big to keep the cylinders visible. Indeed, we choose a radius for the cylinder of $\frac{2\pi}{n}$ with n being the number of attributes on the level. If n is too big, the radius of the cylinder will be too small and we will not be able to see these cylinders. The number chosen here is 500.

But if we want to represent all the attributes of the gene data set, we then need 45 levels which will be displayed one above the other. If we want to represent 45 levels, the representation will not be clear. Also, we will have a very big height on which the different levels will be. We need to keep a certain distance between each level for them to be clearly visible. We then choose to use some reference axes on which the different circle will be displayed to keep a smaller number of levels on each reference axis.

4.2.4 Representation of the levels on the reference axis

We then need to fix the maximum number of levels that will be displayed on each reference axis : the number is fixed to 12, which means that we will represent 12000 attributes by axis. We will then need 4 reference axes to display all the original attributes.

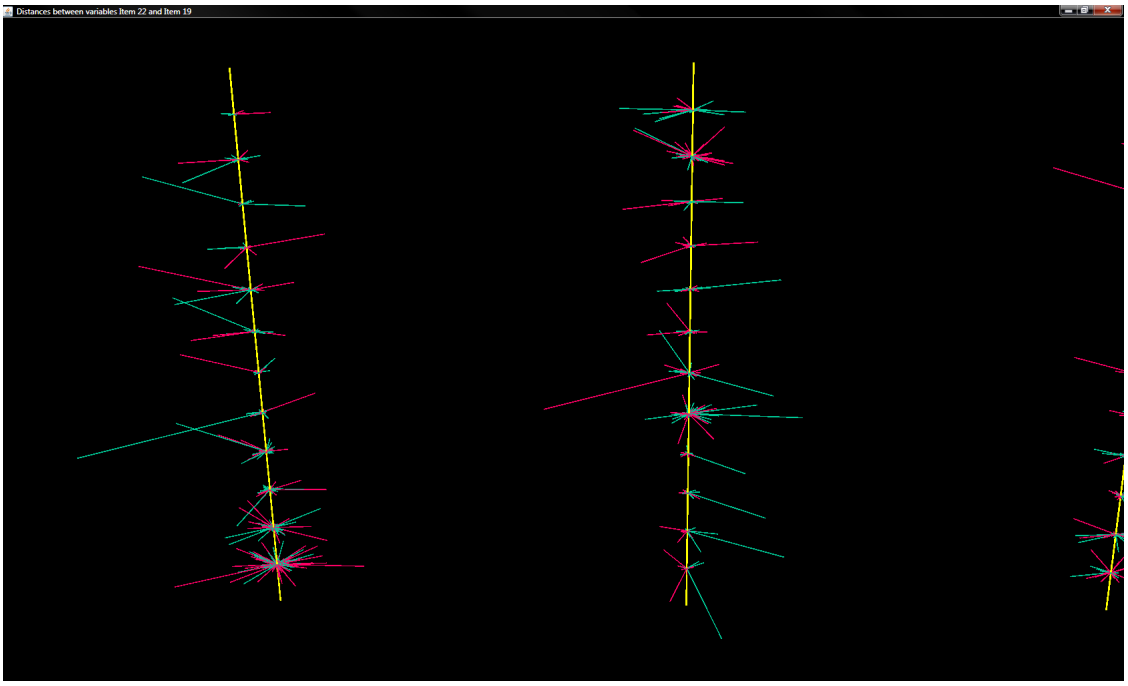


Figure 4.14: Representation of the two first reference axes on the level circles representation

4.2.5 Options applicable on this representation

As we did for the multiple scatter plot, we also added some functions to help the user examining all the levels beside the possibility to navigate inside the visualization.

- **Display the information for each attribute** : When we select one attribute, some information are displayed, like the number of the attribute, the values for each item, the difference between the items and the number of the level or the basis on which this attribute is represented.
- **Selection of a level** : We can also select one level and see this level out of the basis and other levels as illustrated on the following figure :

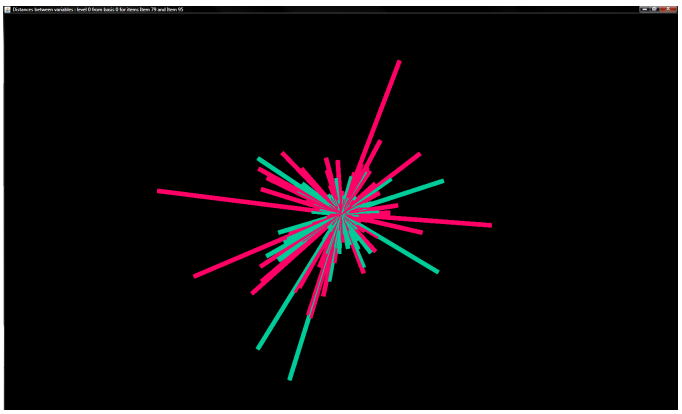


Figure 4.15: Visualization of a level

4.2.6 Discussion

With this visualization it is easy to see when an attribute is very different from one item to another : we simply have a long cylinder. And the attributes that have a small difference have a very small size which makes them nearly invisible. The attention is then carried just on attributes with a great difference.

Furthermore, since an attribute is always at the same place in the representation, it is easy to compare two level circles representation.

The disadvantage of this representation is that we have a lot of information on the same screen (we need to look at the 4 reference axes) but this problem is fixed with the possibility to navigate inside the representation and the possibility to explore one level at a time.

Chapter 5

Exploration

Now that we have the different visualization methods, we can see how to explore the data. The general idea is, from the visualization of the treated data, to help the user going from these treated data to the original data by giving him some indications on what is important, or at least what parts of the data are the most remarkable.

In this chapter, we will see how we can get some general information at one stage of the exploration and how this information can help us to choose the next step. We will also describe the different operations that are applied when we choose one of the different possible options.

5.1 The principles of the exploration

The general principle of the exploration is to start with the groups we built earlier which means starting with the interval data. At the end of this step we already isolated some particular groups : the one that contains only one attribute. These groups are isolated and will not be included in the rest of the exploration. The reason to do so is that we want to extract some particular part of the attributes at each stage of the exploration. Since these groups are very particular, we will not take them into account. These attributes will still be reachable though with the *Select some attributes* option that we will explain later in this chapter.

After the first creation of the groups and the application of the principal component analysis, we start with a multiple scatter plot that shows the results obtained on each item. From this scatter plot and also from different visualizations that we will explain in the first section, we will have the choice between several options : select a principal component, a group or some particular attributes. This will lead us to the next stage of the exploration and to a new multiple scatter plot representing the attributes contained in the principal component or in the groups or the subset of selected attributes.

We will have the opportunity to repeat this selection until we come back to the original attributes. Indeed, we will see that under certain circumstances, we will

be able to get the original attributes. At this stage though, we will have more information about them. The exploration ends with the return to these original attributes.

5.2 Visualizations that help to choose which part to explore

The starting point of the exploration is the multiple scatter plot. From that visualization, we can use several graphics that will help us to decide which part of the data we want to explore. These different graphics are explained in this section.

5.2.1 Scatter plot for the groups

The first graphic is a simple scatter plot of the groups. We represent each group on a scatter plot for which the axes represent the mean and the coefficient of correlation used as references, to build those groups.

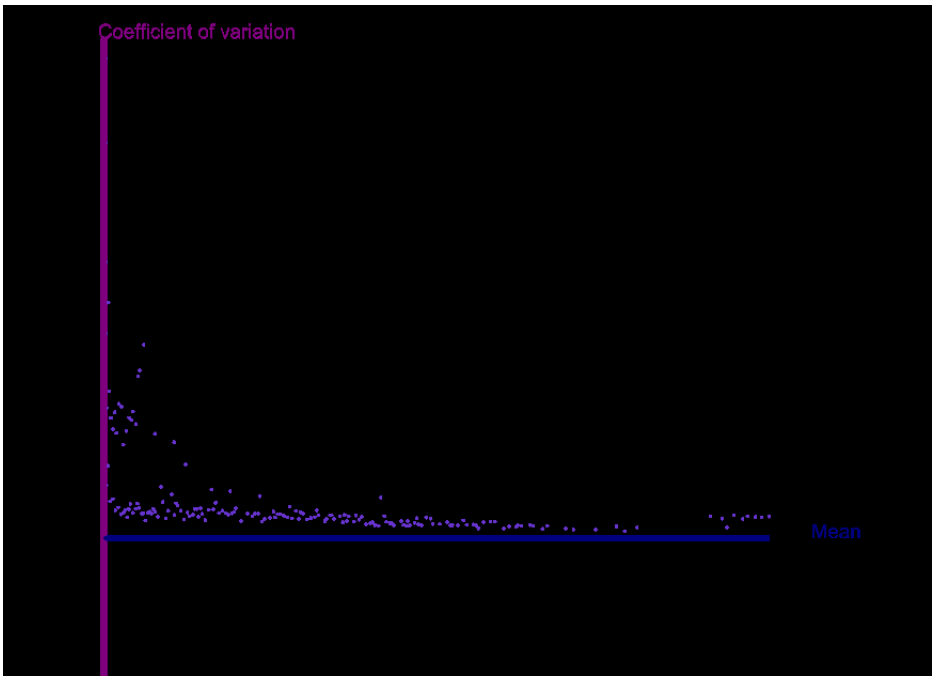


Figure 5.1: Scatter plot for the groups

We also add a functionality on this graphic to give more information about the groups : when we select one of the groups, the following information are displayed :

- the exact reference values for the mean and the coefficient of correlation
- the number of the group(s) with these values (indeed, we can have several groups with the same values, since some groups have been divided with the correlation)
- for each of these groups, the number of elements and the correlation between the elements inside this group (the value 1 means that there is a positive

correlation between the attributes inside the group, the value -1 means that there is a negative correlation and the value 0 means that the attributes are not correlated)

5.2.2 Information on the groups for each principal component

Also, to get an idea of the importance of the influence of the values of each group (or more precisely, the values of the interval data representing this group) on the principal components, we created a graphic that show this influence. This is the coefficient of that group on the principal component. Indeed, each principal component is a linear combination of the original attributes on which we apply the principal component analysis (the interval attributes that represent the groups in our case). The coefficient of each group then reflects the importance of the contribution of this group on the value of the principal component.

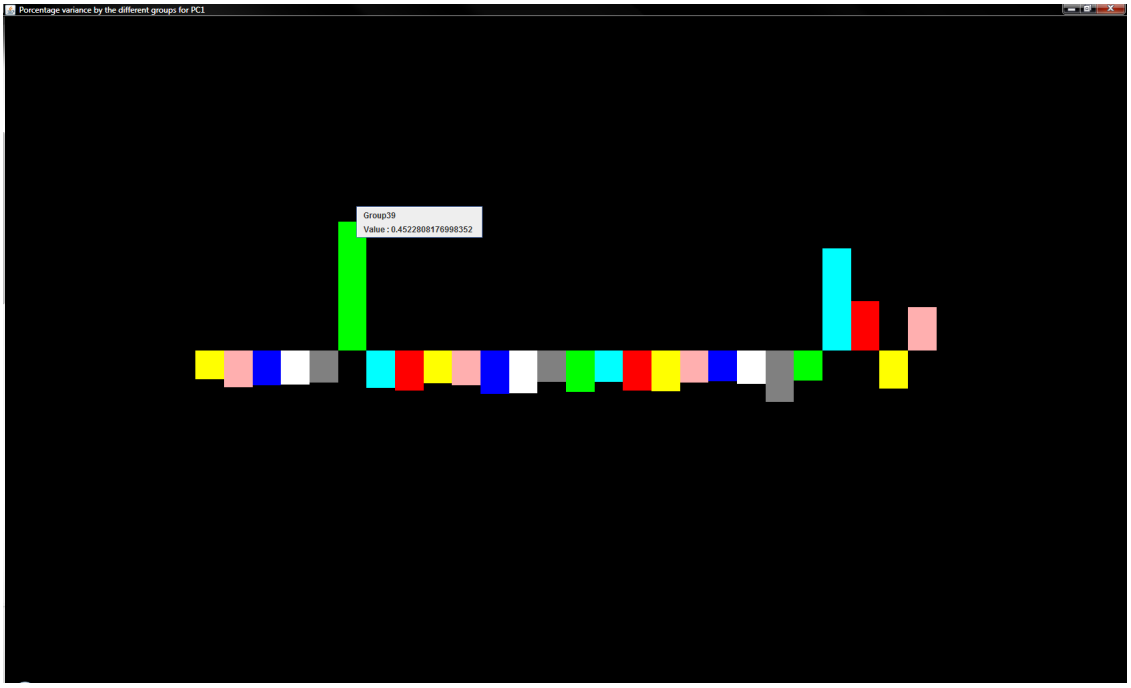


Figure 5.2: Influence of the groups on one principal component

This graphic exists for each principal component. Here, each group is represented by a rectangle. The height of the rectangle reflects the value of the influence and the orientation of the rectangle reflects the sign of the influence (the rectangle is above if the influence is positive and is below if the influence is negative).

Since each principal component is built from all attributes, we should have all the groups on that graphic. But to keep the representation readable, we choose to rep-

resent only the attributes which have an influence greater than 0.1. If the influence is smaller than this value, we can say that the influence is not very strong and we do not focus on them for the moment.

This graphic thus give us a way to measure the importance of a group on the value of the principal components.

5.2.3 Information on all the principal components

The last graphic that can help us to decide the part of the data we want to explore is a graphic showing the percentage of the original data variation explained by each principal component. This percentage of variation is simply the division of the variance explained by a principal component by the total variance that explained the principal component selected.

Figure 5.3: Percentage of variation explained by each principal component

Now that we showed the different graphics that can help us to decide what will be the next step of the data exploration, we are going to explain the different possibilities for the next step of the exploration, and what happens in each of these cases.

5.3 The choice of the next step of the exploration

We already saw that, in order to explore the data, we could choose between three different options. We can :

- select one of the principal component
- select one group

- select a subset of the original attributes

These three options will be detailed in the following sections.

5.3.1 The selection of a principal component

When we select a principal component, we select the groups that have an influence greater than 0.1 on this principal component. Indeed, if we wanted to choose all the attributes that had an influence on the selected principal component, it would mean that we needed to take all the groups. That would be useless. We then need to select the group that had a significant influence on the principal component, and this "significant influence" is fixed to a value greater than 0.1.

This value had been chosen after some tests. Indeed, we wanted to select, at this stage, a number of groups which was acceptable but not too small either. This value seemed to be a correct threshold in order to have a reasonable number of selected groups.

By selecting the groups, we select all the attributes that are inside these groups.

At this stage, we have two cases to examine : if the number of groups with an influence greater than 0.1 is smaller or greater than 10. This value of 10 being the maximum number of attributes that we found reasonable to show on a multiple scatter plot. Since each step of the exploration starts with a new multiple scatter plot, we then have to see if we can represent the groups (or more precisely, the interval attributes representing that groups) immediately on a multiple scatter plot or not.

If there are less than 10 groups :

In this case, we can directly show the interval attributes that represent those groups. Indeed, we can easily show 10 interval attributes on a multiple scatter plot. We then obtain a new multiple scatter plot from these values.

If there are more than 10 groups :

If we have more than 10 groups, we will simply apply the principal component analysis another time on these groups. After this, we visualize the results of this second principal component analysis on a multiple scatter plot.

5.3.2 The selection of a group

When we select a group, we select all the attributes contained inside this group. As in the previous case, we will have two different cases : if there are less than 10 attributes in the group or more than 10 attributes. The treatment of the group will

be different in both cases.

If there are less than 10 attributes in the group :

If we have less than 10 attributes, we can show directly these attributes on a multiple scatter plot. But this time, we will work on the original attributes. That is the way we will come back from the interval attributes we built with the groups to the original data.

If there are more than 10 attributes in the group :

If we have more than 10 attributes inside the group, we will apply the method of grouping once again on these attributes. But this time we will use a smaller percentage of variation of the mean, in order to have thinner groups. We will not change the percentage of variation. The idea behind this way of doing things is that if a group contains a lot of attributes, maybe we will have an interest in creating some other groups in order to have less attributes in each groups. The attributes that are in the same group will then be closer one from the other than the attributes in the original group are.

Each time, we will use a coefficient of variation of the mean, which is the half of the value use to create the group we want to explore.

Then, we will build new interval data. Before starting the visualization we also need to examine the number of groups that we are creating from this group. Il the number of groups is smaller than 10, we can directly show the groups on a multiple scatter plot, exactly as if we select a principal component and that the number of significant groups in this principal component is smaller than 10.

However, if the number of groups is greater than 10, we will do the exact same thing than in the very first phase : we will apply the principal component analysis on the new groups, and the results of the principal component analysis will be represented on a multiple scatter plot. One important thing to notice is that we only show the groups that contain more than one attribute as we explained sooner in the introduction of this chapter.

5.3.3 The selection of some attributes

We can also select a subset of attributes that we have noticed in one of the stage. This can be used for example for the attributes that have been isolated in a group that contained just that attribute. Or also, we can select some attributes that we knew before the exploration for example.

Chapter 6

An example of exploration

In this chapter we will give an example of how we can explore the data to illustrate how we can use the different visualizations and how to interpret them.

We will just describe one step in the exploration since each step can be realized in the same way except in some cases that we will describe here.

6.1 First step of the exploration

We start the exploration from the first multiple scatter plot that represents the 9 principal components we built on the first groups created on the entire data set. From this scatter plot we can make different things : explore this multiple scatter plot, focus on some items, see the differences between some objects. We can also see the different graphics that can help us to choose the next step of the exploration.

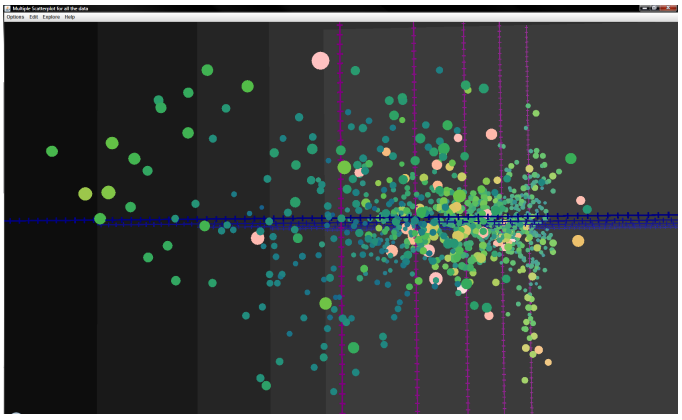


Figure 6.1: First multiple scatter plot

6.1.1 Exploration of the multiple scatter plot

We start the exploration of the data set with the first multiple scatter plot. Since it is the very first visualization and that this visualization is built on all groups, the information displayed is very general. Nevertheless it allows us to have a general idea of how the points are displayed on the different scatter plots and to compare the general shape of the set of points. For example, we can see on which scatter plot, and thus on which principal component, the points have a smaller or bigger value or again, on which scatter plot they are more concentrated or not. We can start, for example, by selecting one of the scatter plot in order to get a more precise idea of the information we can find on this scatter plot.

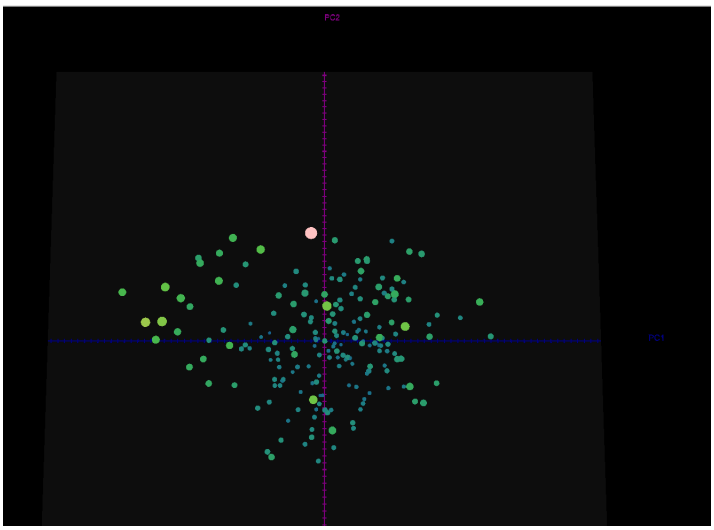


Figure 6.2: First scatter plot from the multiple scatter plot representing the two first principal components

From this scatter plot, we can notice that all the items are approximately grouped. We do not find any outliers. But we can also see that there is two different groups in the points : one is formed with small points in a blue color and the other one is formed with points a little bit bigger and in a green color.

One noticeable point is bigger than the others but also it has a different color : it is in a pink color (item 6). It is the only one with these characteristics. We can then compare it with some other points in order to get an idea of the attributes that could explain this difference. We will use the visualization of the distances in order to do this. For this, we will compare this object with another object, a small blue point (item 77) :

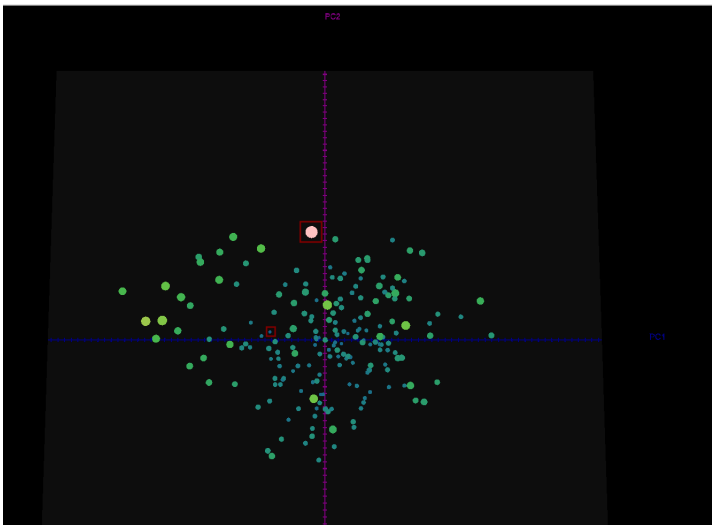


Figure 6.3: Selection of the two items we will compare

But before going to the distances visualization, we can also simply examine the exact values for these two items :

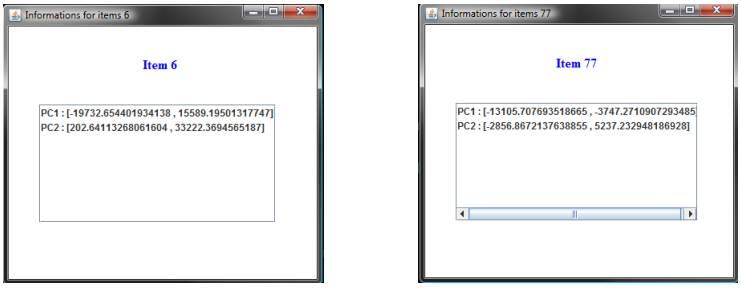


Figure 6.4: Exact values for the two selected items

From these values, we can say, for the first principal component, that the span of the interval is indeed really different for both objects (around 35000 for the item 6 and only 10000 for the item 77). For the second principal component, the pink color corresponds to a big span of interval (around 33000) and the blue color to a smaller one (around 8000).

We then have an item with a big span for both intervals (the item 6) and another one with smaller values for both (item 77). Also, if we look at their position on the scatter plot, we can notice that they are centered at a not so different place for the first principal component but for the second principal component, their positions are very different.

6.1.2 Focus on some items and distances between them

To see the distances, we can use the level circles representation. On this representation, we can see that some attributes are really different for the two items. Maybe these attributes will be interesting to explore in a separate step as we will see later in this chapter.

We can see here the interest of this visualization : the differences for all the original attributes are shown here but, as we can see on the level circles representation, we do not have 22280 visible attributes. Only the attributes with a noticeable difference show up.

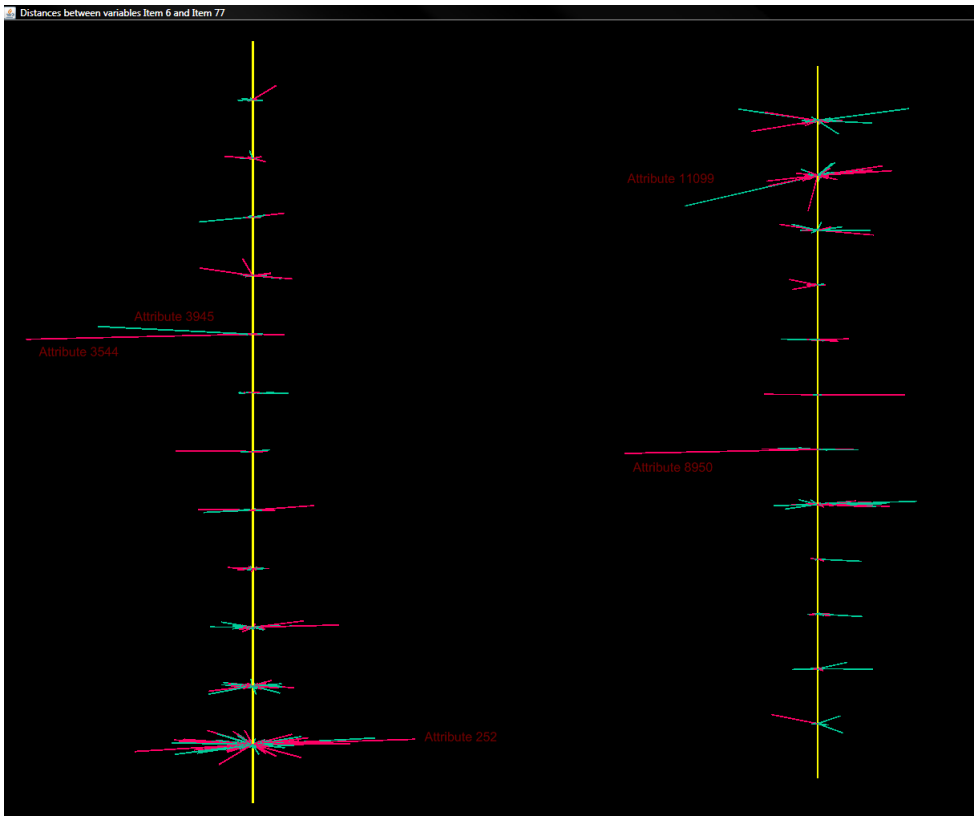


Figure 6.5: Distances representation for the 12000 first attributes

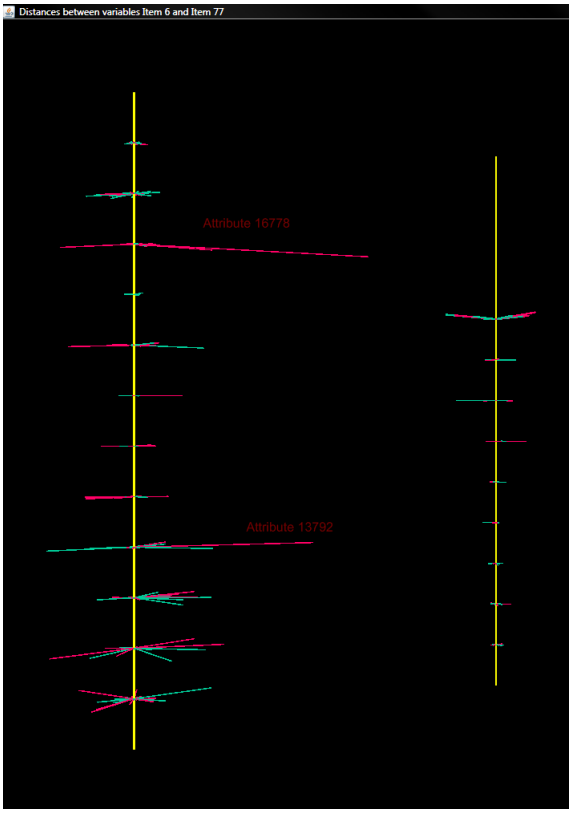


Figure 6.6: Distances representation for the 10280 last attributes

We can also go further with this representation with an exploration of the different levels. For example, for the very first level, we can see it on figure 6.7.

With the help of this representation of a level, we can see with more precision the difference for all the attributes because, from the original representation of the level circles representation, the biggest differences are only noticeable.

6.1.3 Focus on one item on all the scatter plots

To get more information about the items, we can also compare its position, size and color on each scatter plot as shown on figure 6.8.

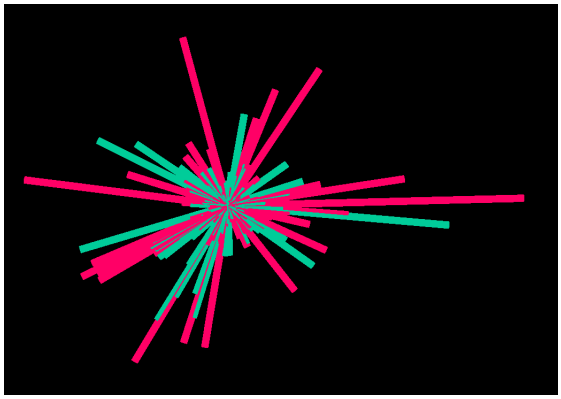


Figure 6.7: Representation of the first level of the first basis

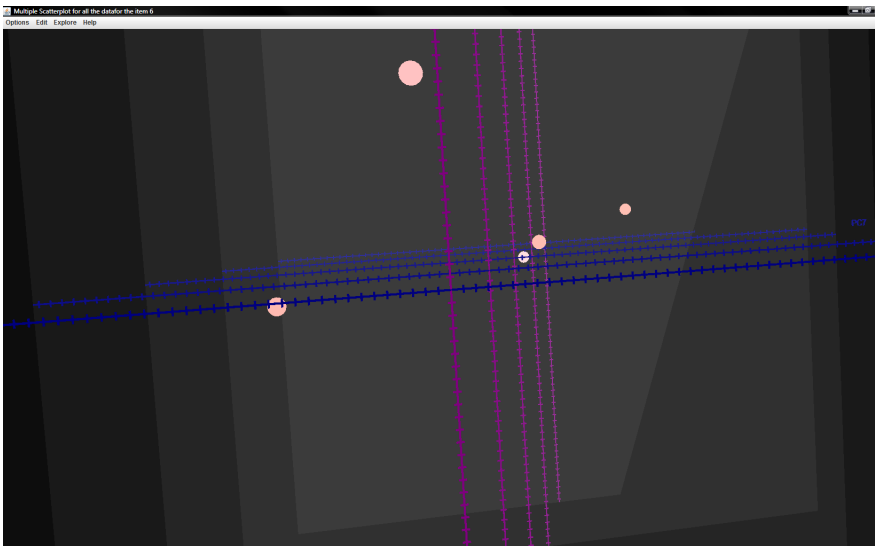


Figure 6.8: Representation of the sixth item on all the scatter plots

This way, we can notice that this particular item stay in the same color on every scatter plot (indicating that the value it takes is similar for each y-axis on each scatter plot). But its size seems to decrease from one scatter plot to the next one indicating a decreasing value for the principal component showed on the x-axis.

But we need to be carefull with the interpretation : the value of the second axis may be decreasing too. We may just not notice it because the value decreased not so fast and thus stay in the same color on the scale of colors. But to be sure of that observation, we can simply examine the exact values for each point.

We can also use the differents graphics to have more information on the attributes and to help us to choose one of the next possible steps in the data exploration.

6.1.4 Getting some information on the attributes

From some additionnal graphics, we can have some information about the attributes represented on the multiple scatter plot and also about the different groups. With this information we will be able to choose the next step we want to execute.

The percentage of variation explained by each principal component :

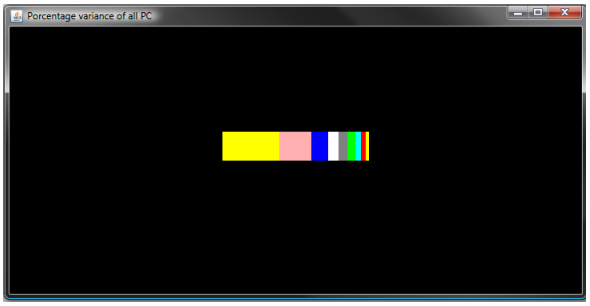


Figure 6.9: The variation explained by each principal components

From this graphic, we can learn that the first principal component explains 38.9% of the variation of the original data. This principal component is then very important.

The influence of the groups on each principal component :

We can also see which groups are the most influent on each principal component and get the value of this influence. For example, for the first principal component we have :

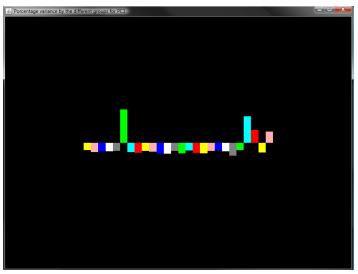


Figure 6.10: Influence of the groups on the first principal component

From this graphic, we learn that two groups have a very big influence on the values of this principal component. The group 39 has an influence of 0.45 and contains 312 attributes of the original data. The group 133 has an influence of 0.39 and contains 5 attributes.

From these observations, we can now select one of the next possible steps of the exploration.

6.1.5 Selection of a principal component

We can select the first principal component since it explains nearly the half of the variation in the original data set. We can then select this principal component and obtain a new multiple scatter plot. This new multiple scatter plot will be the starting point of the second step in the exploration.

When we select this principal component, we select the 26 groups that have an influence bigger than 0.1. Since the number of significant groups is greater than 10, we apply the principal component analysis a second time. Three new principal components are then built. These results are shown on the following multiple scatter plot :

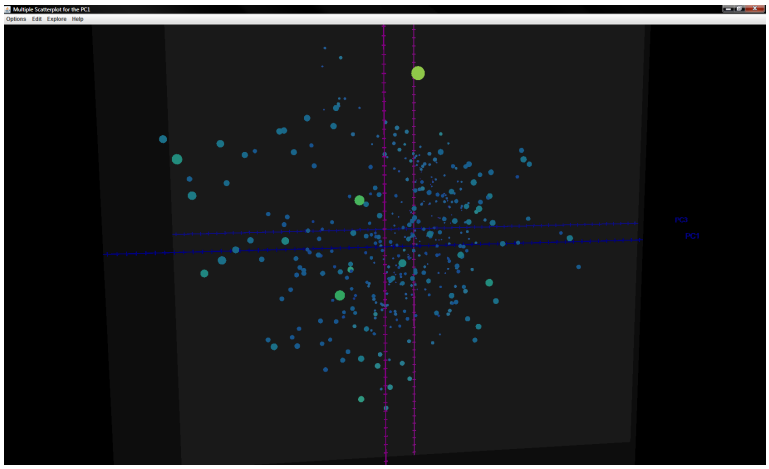


Figure 6.11: Multiple scatter plot obtained once we select the first principal component

6.1.6 Selection of a group

Since the first principal component is more important and that this principal component is mostly influenced by the groups 39 and 133 we can also think that we could select one of those two groups.

The selection of the group 39 :

If we select this group, containing 312 attributes, new groups will be built : we will have 72 new groups, 20 of them containing just one attribute. The principal component analysis will then be applied on these groups. We will obtain 7 new principal components represented on the following multiple scatter plot :

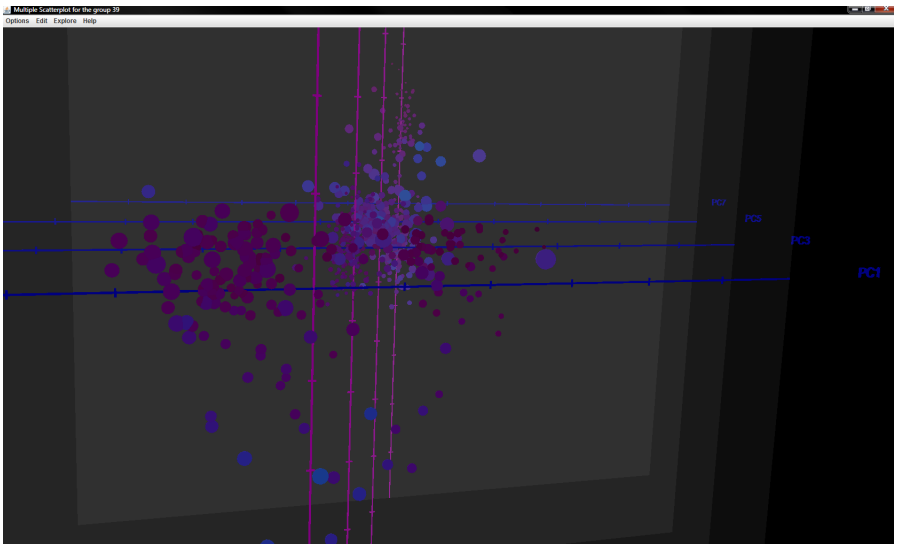


Figure 6.12: Multiple scatter plot obtained once we select the group 39

The selection of the group 133 :

If we select that group containing only 5 attributes, these attributes will immediately be displayed on the multiple scatter plot.

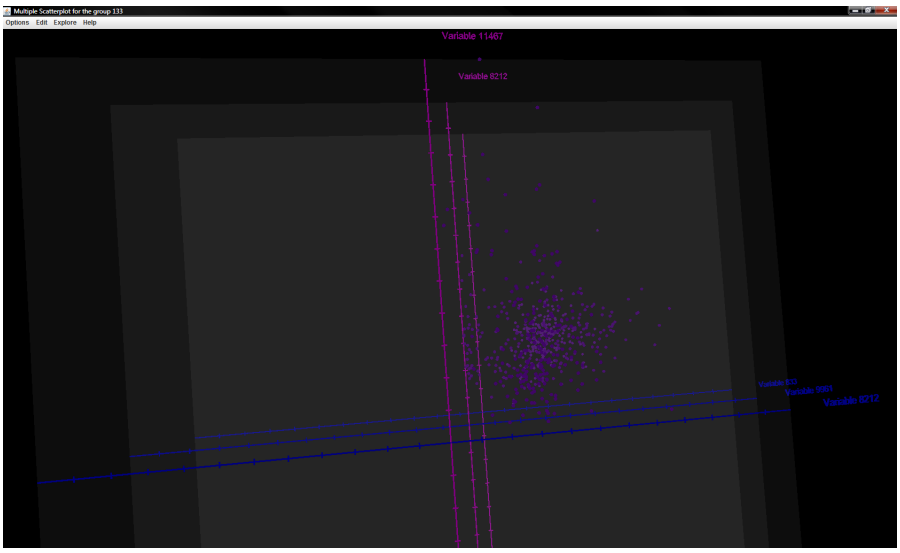


Figure 6.13: Multiple scatter plot obtained once we select the group 133

Another way to find a group to explore is to look at the scatter plot of the groups :

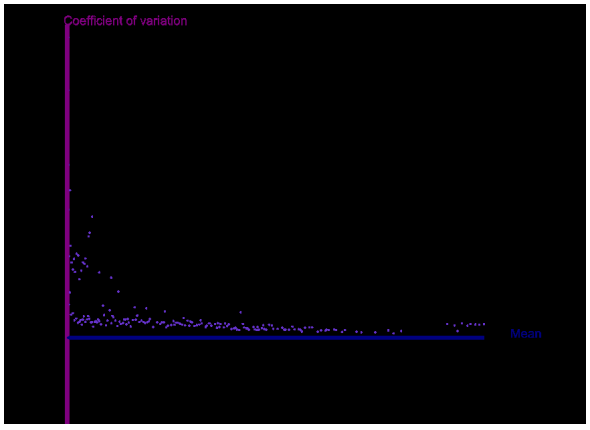


Figure 6.14: Scatter plot of the groups

From this scatter plot, we can notice that the majority of the groups have a low value for the coefficient of variation. But some of them have a high value. Maybe we could explore them. Also, we can notice that there are some groups that have a bigger value for the mean. Maybe these groups would be interesting to explore too.

6.1.7 Selection of a subset of attributes

This option can be used to represent, for example, the attributes that we found very different from one item to another on the representation of the distances.

Also, when we created the first version of the group, we had a list of attributes that were all assigned to a different group meaning that these attributes were more different from the other one. We can then think about taking some of these attributes and explore them.

We can, for example, select the attributes 8223 and 7303 that were isolated at the end of the first groups construction or these two attributes and a third one, 16598.

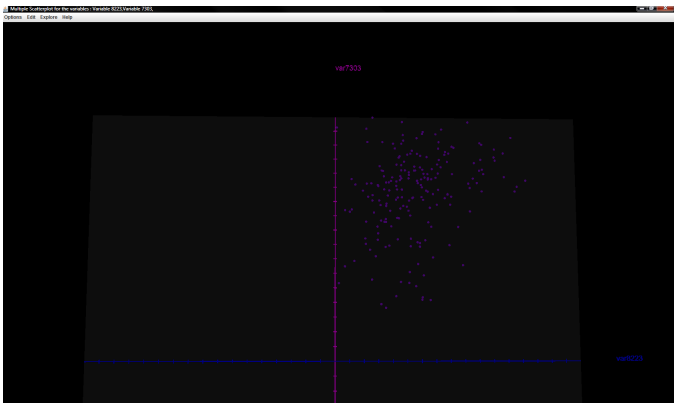


Figure 6.15: Selection of 2 attributes

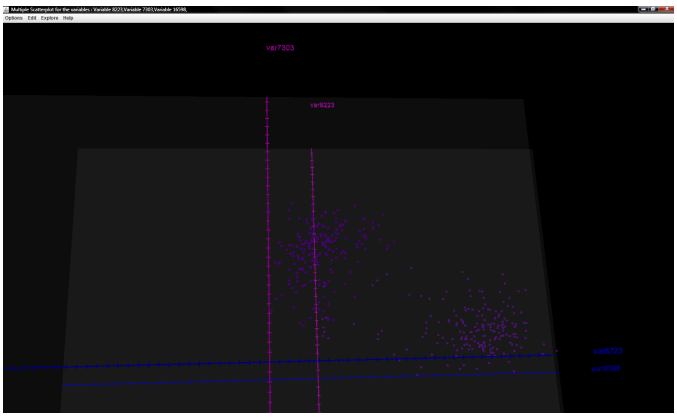


Figure 6.16: Selection of 3 attributes

6.2 Second step of the exploration

For the second step of the exploration, we have different options depending on the attributes (eventually contained in the group or in the principal component we choose) we select during the previous step. Following the number of attributes, we can obtain some new principal components or some new groups but we can also find back some of the original attributes.

If we have some new groups or principal component, we simply repeat the previous step. We explore them, get some information and select some attributes for a third step. But we can also think that the direction we took was not interesting finally and we can choose to come back to a previous step, and select another group or principal component to explore.

If we find back some original attributes, we can not go further in the exploration. This is the final step of the exploration. When we arrive at this stage, we can continue to explore some other previous groups or principal components.

Chapter 7

Conclusion

We described a tool that allows us to visualize and explore the gene data set which is characterized by a huge number of attributes. The visualization implied the necessity to decrease the number of attributes. This was done by the mean of a grouping method, and the application of the principal component analysis.

Once the number of attributes is usable by some visualization method, we can start the data exploration with a generalization of the classical scatter plot, the multiple scatter plot. We can explore the data with the help of this graphic but also with some other graphics that give us some information about the structure of the principal components and the groups. This can be done by choosing to explore a principal component or a group or a set of attributes successively. This way of doing things allows us to go from the new built interval data to the original attributes without having to search some attributes of interest among the 22280 original attributes.

7.1 Discussion

We can notice that the visualizations developed here are very clear and can be understood by people that are not mathematicians or computer scientists. But this simple representation increase the time needed to realize a complete exploration of the data set. Even if it is not really feasible to find a representation of a data set containing so many attributes that will be really straight, the time needed to explore all attributes is quite long.

The advantage that we can find with this representation is that some attributes might be discarded, since we may notice in one stage that they are not really significant for the data set. This way, we probably do not have to examine each attribute.

The tool developed here is very general and a lot of points could be optimized in order to take into account the meaning of the data. We can say that we have a general tool that had to be parameterized for the data set we want to work on. We could for example, change the parameters we used for the construction of the groups (the mean, the coefficient of variation and the correlation). Indeed, depending on

the meaning of the data, we could prefer to take into account some other parameters (for example, maybe the mean does not matter but the variation of the values of this interval is more significant). We could also optimize some other parameters as we discussed in the previous chapters.

After a presentation of the tool to Daniel Catchpoole, a biologist who works at the Children's hospital at Westmead, from where the data came from, it seemed that the multiple scatter plot was a useful representation and that the maximum number of variables we represent on it (10) was not too important, and could be easily understood. For the representation of the distances on different levels tough, some additional work had to be done. Indeed, the visualization of every attribute in the data set may not be very useful. We could think about representing a general distance for the groups instead of each attribute or adapting the distance used to take into account the real meaning of the data.

7.2 Future work

The first thing we could do is to precise the exact meaning of the gene data set. The tool could then be optimized and configured for this data set, but we could also add some visualizations to express some specificities of this data and some biological knowledge. Following this idea, we can also think to use some other versions of the data created after some different treatments by the Children's cancer research unit from the Children's hospital at Westead. We could then adapt the part of the tool dealing with the data, before using the visualization methods. We could then try to find better results for this treatment and thus for the data used by the visualization part of the tool.

We could also add a part to the tool that would give a description of the groups and that would extract some information directly from the attributes included in the different groups. For example, we could develop some visualization methods allowing to see the repartition of the attributes around the values of references taken to create the groups).

On another hand, we could also introduce some additional visualization, to represent the different steps of the exploration and to express more clearly what happens between two steps.

Another point we could improve is the interaction. Indeed, for the moment, the only interactions we introduced was the possibility to navigate inside the representations, selecting some particular parts on which we wanted to focus, and the possibility to select the set of attributes (the groups or the principal component) that would be used on the next step of the exploration. We could add some additional options to allow the user to have more interactions with the visualizations and also between the different stages of the exploration.

The tool was developed on the attributes side and we could also try to include some more precise information about the items, which means the patients. We could continue to develop the tool that way and include some methods to analyze and compare one or more patients. We could, for example, try to find some classes to identify some particularities of the patients and these classes could help us to have some additional information if we added a new patient on the data set.

We can then say that the tool we developed here is a basic tool, but that it can be completed and improved before being really used in a medical context.

□

Bibliography

[1] Springer series in statistics : Modern multidimensional scaling. http://www.springerlink.com/content/x34xtt6q85l39t16/. Springer New York, Second Edition.

[2] M.l Ankerst. Visual data mining with pixel-oriented visualization techniques. citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.134.2511.

[3] Answers.com. Oligonucleotide. http://www.answers.com/topic/oligonucleotide.

[4] K. Baumann. Submission to rhizome's track 1 : alt.interface. http://www.artifar.com/submission/pick.html.

[5] B. L. BOUJELOUD-ASSALA. *Visualisation et algorithmes genetiques pour la fouille de grands ensembles de donnees.* PhD thesis, Laboratoire d'Informatique de Nantes Atlantique, 2005.

[6] C. Pellillo Brase C. Henry Brase. *Understanding basic statistics.* Brief ed, 2008.

[7] Wikimedia Commons. File : Color icon pink.svg. http://commons.wikimedia.org/wiki/File:Coloricon -pink.svg.

[8] J.-D. Fekete C. Gorg J. Kolhammer G. Melanĉon D.l Keim, G. Andrienko. Visual analytics' : Definition, process, and challenges. LNCS 4950. 2008, pp 154-175.

[9] M. Noirhomme-Fraiture E. Diday. *Symbolic data analysis and the SODAS software.* Wiley-Blackwell, West Sussex, 2008.

[10] M. Friendly. Milestones in the history of thematic cartography, statistical graphics and data visualization. http://www.math.yorku.ca/SCS/Gallery/milestone/milestone.pdf. August 2009.

[11] E. Diday H.-H Bock. *Analysis of symbolic data : exploratory methods for extracting statistical information from complex data.* Springer Edition, 2000.

[12] S. Kaski J. Venna. Comparison of visualization methods for an atlas of gene expression data sets. *Online publication*, May 2007.

[13] K.A. Cook J.J. Thomas. *Illuminating the path.* IEEE Computer Society Press, Los Alamitos, 2005.

[14] D. A. Keim and Ieee Computer societyl. Information visualization and visual data mining. *IEEE Transactions on visualization and Computer Graphics*, 8:1–8, 2002.

[15] Electronic Visualization laboratory. Parallel coordinates. http://www.evl.uic.edu/aej/526/kyoung/Training-parallelcoordinate.html.

[16] Dr. Y. Liu. Visualization of multivariate data. http://www.engineering.wright.edu/ yan.liu/IHE631/MultivariateDataVisualization.pdf.

[17] H.-P. Kriegel M. Ankerst, D. A.Keim. 'circle segments' : A technique for visually exploring large multidimensional data sets. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.68.1811. Institute for Computer Science, University of Munich.

[18] P. Niyogi M. Belkin. Laplacian eigenmaps for dimensionality reduction and data representation. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.131.3745. December 24, 2001.

[19] H. Schumann M. Luboschik. Explode to explain. http://www.infovis-wiki.net/index.php?title=ExplodetoExplain.

[20] M.Noirhomme-Fraiture. Visualization of large data sets : the zoom star solution. Journal of symbolic data analysis, vol.1. July 2002.

[21] A. Nahimana M.Noirhomme-Fraiture. Multimedia support for complex multidimensional data mining. workshop on symbolic data analysis, inside PKDD'2000. 2000.

[22] C. Mazel M.Noirhomme-Fraiture, A. Nahimana. Temporal symbolic description graphics in iso-3d. workshop 'multimedia data mining', inside KDD'2000. September 2000.

[23] M. Noirhomme-Fraiture. Multimedia support for complex multidimensional data mining. *Workshop on Multimedia Data Mining*, pages 54–59, 2000.

[24] B. Otjacques. *Techniques de visualisation des informations associées 'a une plate-forme de coopération.* PhD thesis, Facultés universitaires Notre-Dame de la Paix - Namur, 2008.

[25] ASSO Project. Sodas2 software user manual. http://www.info.fundp.ac.be/asso/sodaslink.html.

[26] S. Juhash S. Kromesch. High dimensional data visualization. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.108.3671. Budapest University of Technology and Economics.

[27] L. K. Saul S. T.Rowei. Nonlinear dimensionality reduction by locally linear embedding. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.111.3313. November 2000.

[28] SAS. Example 1 : Creating a scatter plot matrix. http://support.sas.com/documentation/cdl/en/grstatproc/61948/HTML/default/a003155769.

[29] Ulg. Chapitre 7 : 3. coefficient de corrélation. http://www.astro.ulg.ac.be/cours/magain/stat/stat73.html.

[30] T. Walsh. *Dimensional Stacking in Three Dimensions*. PhD thesis, Worcester Polytechnic Institute, 2008.

Annexes

Chapter 8

Manual for the tool

Introduction

The tool developed to visualize the gene data set consists in two programs. The first one aims at creating some groups in the originals attributes and to apply the principal components analysis on these groups. This way, we build three different files that are used as input by the second part.

The second program creates a first multiple scatter plot and from this scatter plot, we can visualize some particular parts of the data and explore it.

First step : Construction of the groups and of the interval principal components

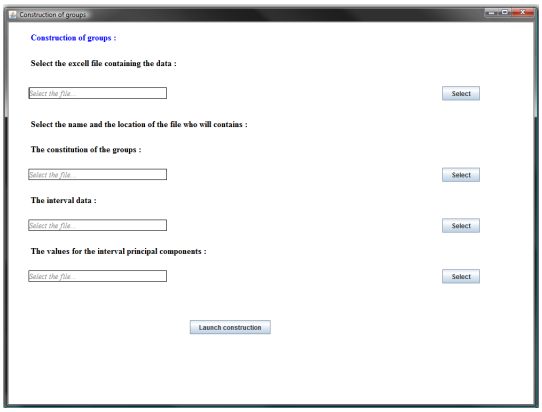
This first program is in the jar file *ConstructionGroupsAndPCAresults.jar*. To launch it, you need to use the command : *java -jar -Xmx1024m ConstructionGroupsAndPCAresults.jar*. The part *-Xmx1024m* in the command line is to indicate to the java machine that the program needs 1024 Mo to run.

Once launched, you have the following window :

To launch the construction of the groups and of the principal components, you need to specify the excel file that contains the data. Here, we use the file *Combine_Expresso.xls*.

You also need to choose the files that will contain :

- The different information about the groups that are built
- The values for the interval data (built from the original data)
- The values of the interval principal components



These three files have to be text files (.txt extension). They will be used by the visualization program.

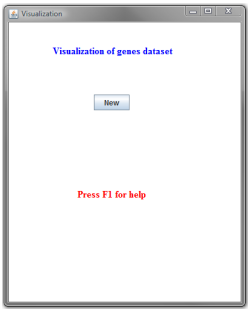
Once you gave those choices, you can launch the construction. The construction is quite long (about an hour) but it is not necessary to perform it each time you want to use the visualization program. You just need to do this part again if you change the data file. That is the reason why this construction is done by a different program.

At the end of the construction, you will have the three files containing all the information you need to visualize the data.

Second step : Visualization and exploration of the data

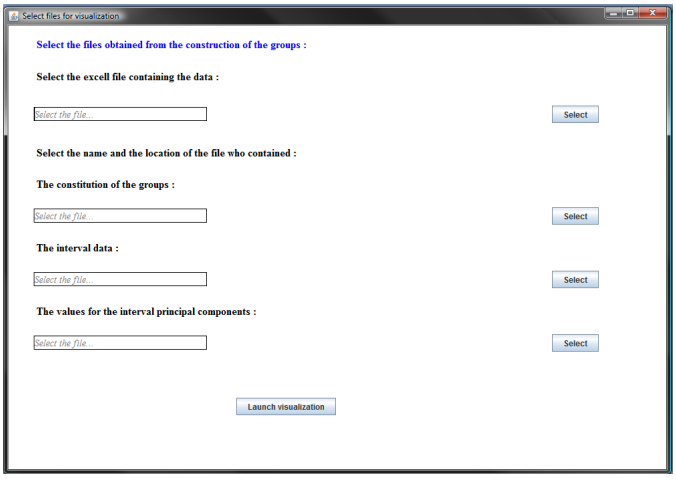
This program is in the jar file *Visualization.jar*. To launch it, you need to use the same command than before : *java -jar -Xmx1024m Visualization.jar*.

Once it is done you have the following window :



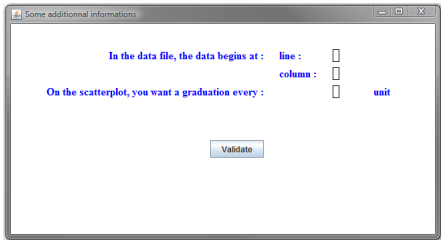
New exploration/visualization

You have to specify the location of the data file and of the three files built with the construction program :



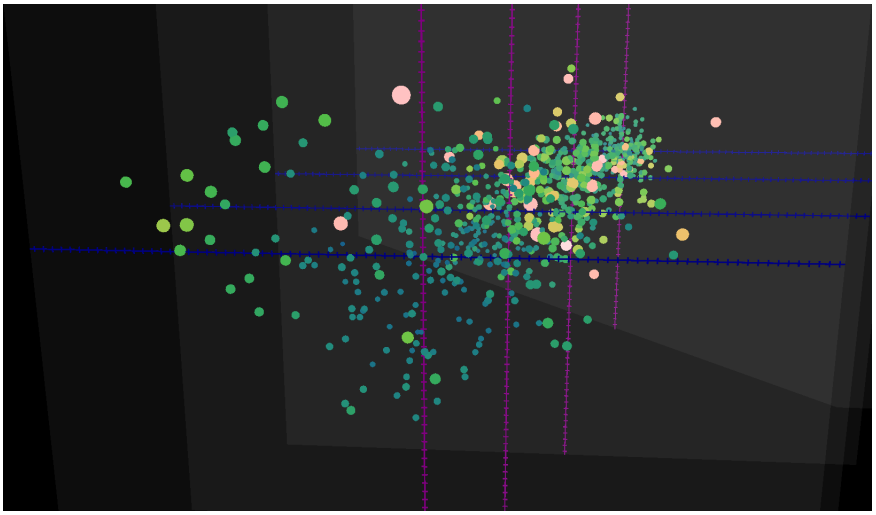
Before starting the visualization, you still have to give three information :

- The number of the first line containing the data (not the name of the genes)
- The number of the first column containing the data (not the name of the patients)
- The value between two graduations on the scatter plot.



With the *Combine_Expresso.xls* file, the values are 1 for the line, 1 for the column and with 1000 for the value between two graduations we can have a good idea of the different values on the multiple scatter plot.

And now, you finally have the first multiple scatter plot :



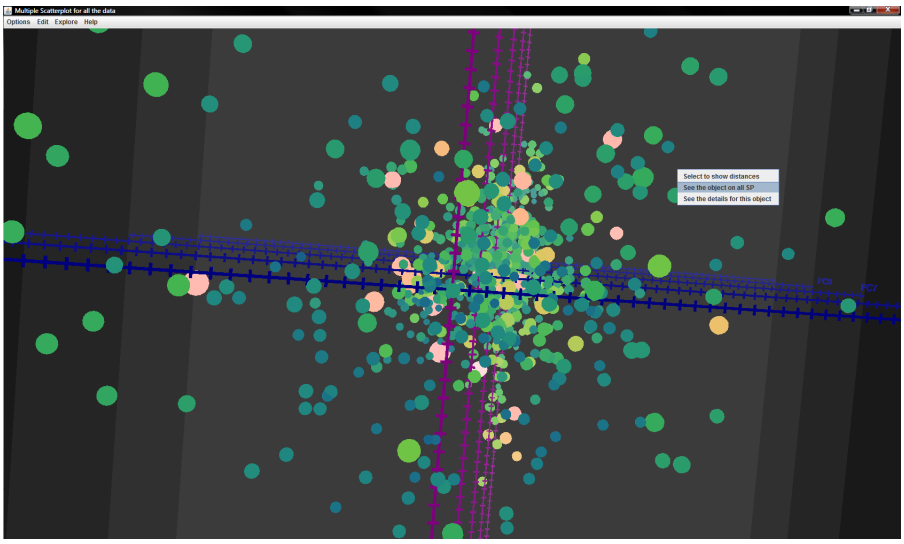
This multiple scatter plot is the starting point of the visualization and of the exploration of the data. All the principal components are shown on this scatter plot. The first one represents the first principal component with the second one, the second scatter plot represents the third and fourth principal component, and so on,... As the number of principal components is odd, the fifth scatter plot shows the ninth principal component with the first one.

On each scatter plot, we find the values for all the patients.

We can now see all the operations and functions we can apply from this basis multiple scatter plot.

Operations on the points of each scatter plot

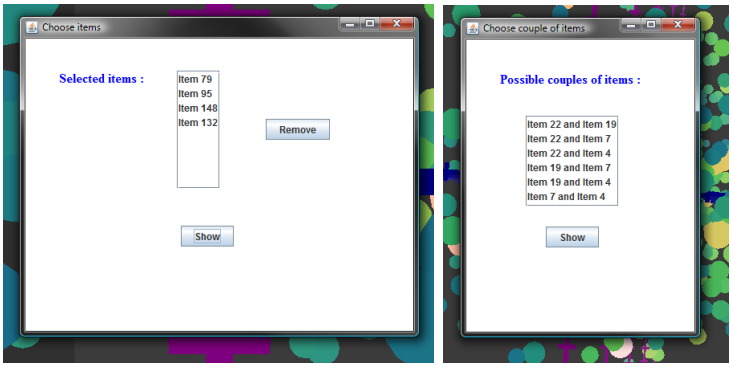
When you select one point on a scatter plot, you have different options :



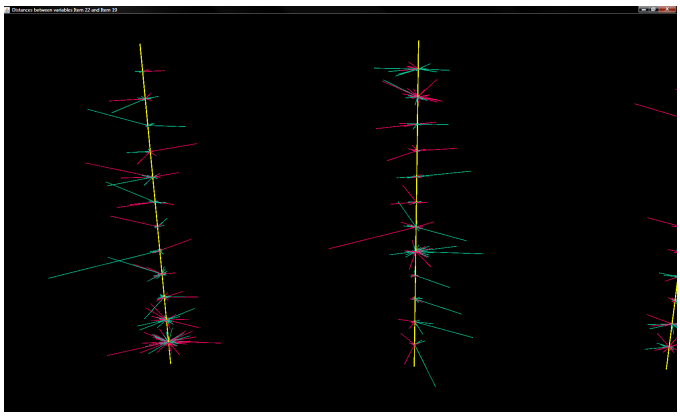
Select to show distances

This option allows us to focus on the distances between two patients. This distance is simply the difference between the values of these items for all the attributes.

You can select as many items as you want, all the possible combinations will be compute and you will be able to choose which one you want to see.



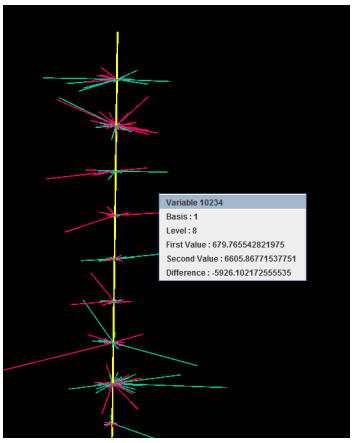
Once you selected a combination of items you want to see, you have the following representation :



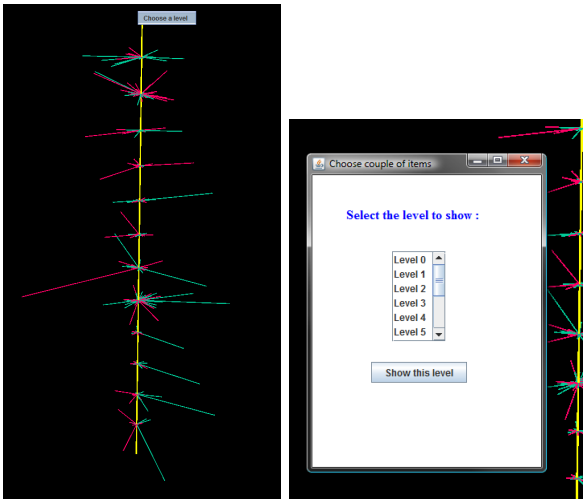
On this representation, the distances between the two selected items for all the attributes of the original data are shown. One cylinder on one level represents one attribute and the length of the cylinder is proportional to the difference between the values for this attribute on the two patients. If a cylinder is long, it means that there is a big difference between their values and if it is short it means that the difference is small.

The different values are divided into several *basis* (yellow cylinder) and in *levels* for reasons of readability. One level represents 1000 attributes of the original data.

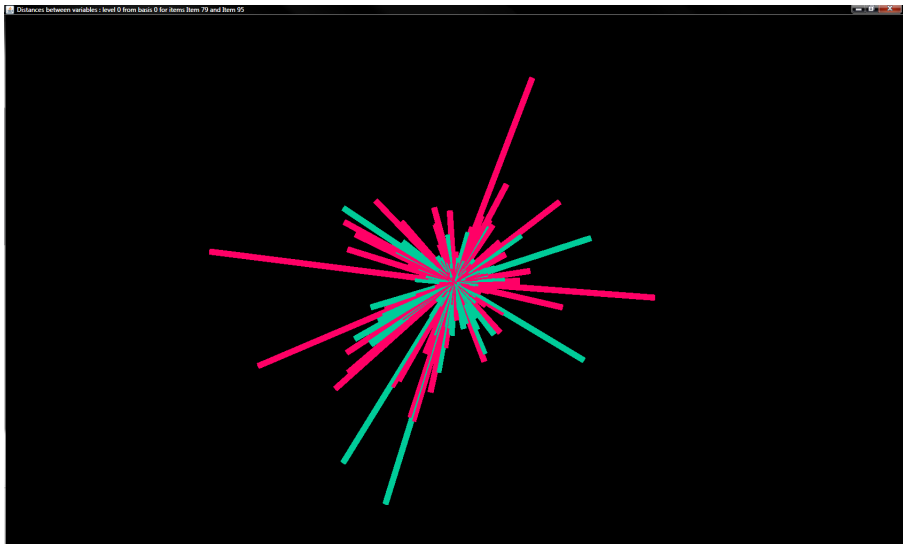
From that representation, you can have the details for each object by clicking on the object :



You can also choose one level to see it more clearly :

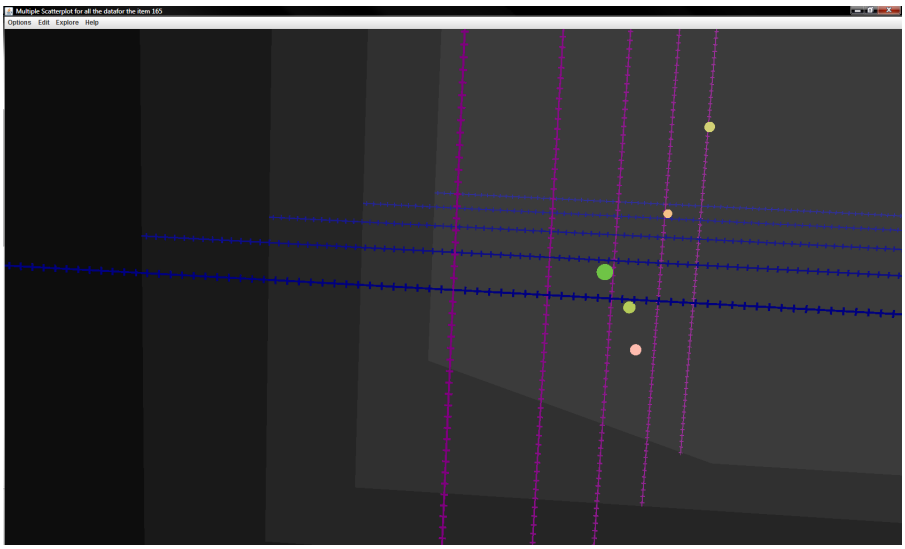


Then you will see just one level :



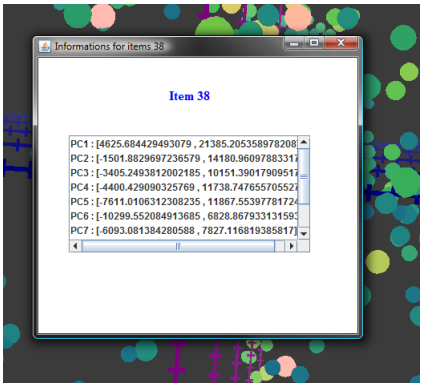
See the object on all SP

From one scatter plot, you can also select just one object that you will see on each scatter plot :



See the details for this object

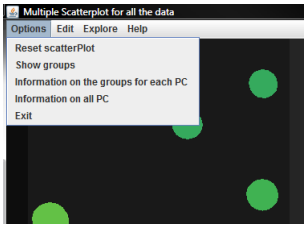
You can also have the exact values for the principal components for one object :



Menu

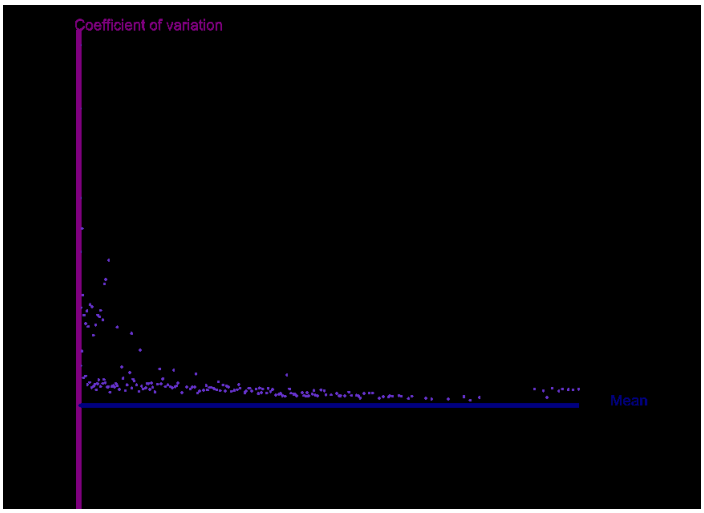
Now that we examined the different options for the patients, we are going to see the functions for the multiple scatter plot and the options that allows us to explore the data by seeing all the options in the menu.

Menu Options



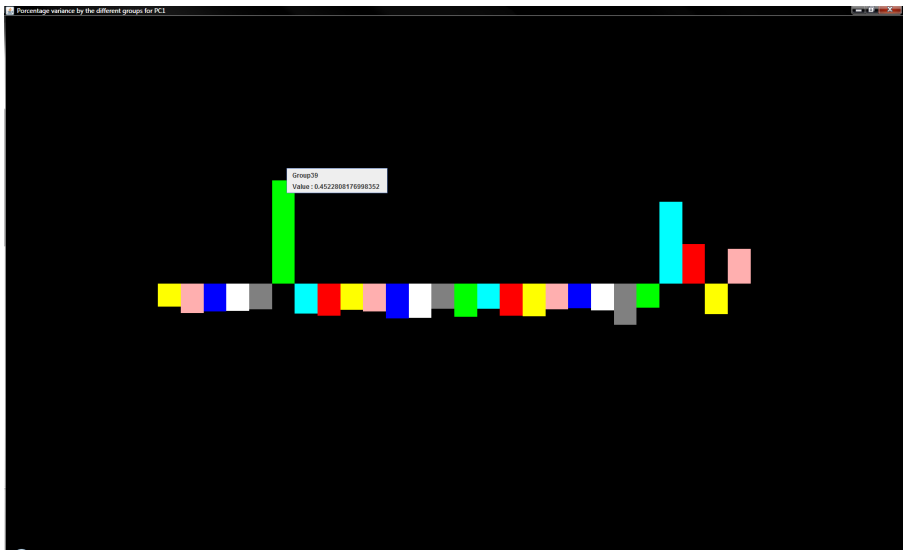
The **reset scatter plot** option allows us to come back to the original multiple scatter plot that we obtained when we launch a new visualization.

The option **show groups** gives us a representation of the groups built on the original data :



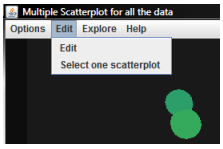
This representation shows the groups on a scatter plot made with the variables mean and coefficient of variation that are the parameters used for the construction. We can also have more information on each group by clicking on it.

The **Information on the groups for each PC** allows us to have, for a selected principal component, a measure of the influence of one group on the values of the principal component. A rectangle which is above means that the group has a positive influence and if it is below it means a negative influence. Once again you can have more information by clicking on a rectangle :



The **information on all PC** option gives us a measure of the percentage of variation in the original data explained by each principal component. If a rectangle, representing a principal component, is bigger it means that this principal component explains a bigger part of the variation.

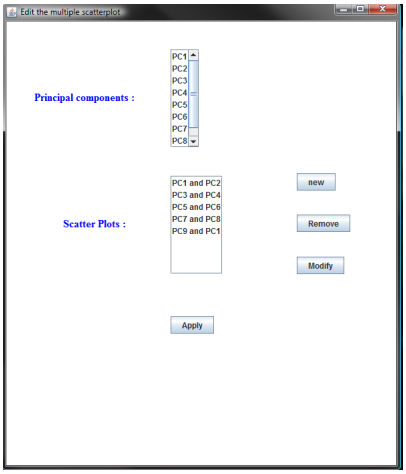
Menu Edit



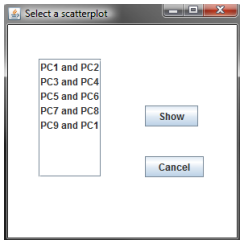
The edit menu contains all the options that allow us to modify the original multiple scatter plot.

The **edit** function give us the possibility to modify the position of the different scatter plots, to modify the association between the principal components on each

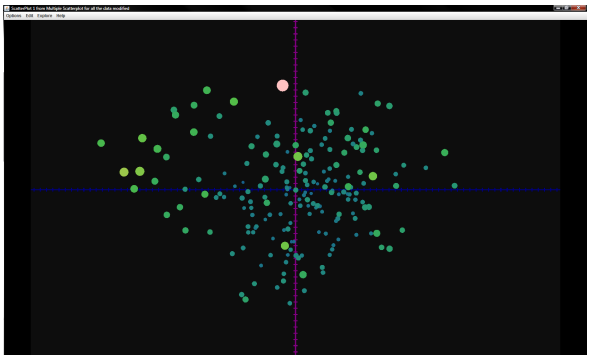
scatter plot and to add or delete a scatter plot :



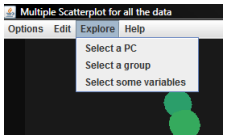
The **select one scatter plot** option gives us the possibility to show just one particular scatter plot of the multiple scatter plot :



This way, we can see each scatter plot more clearly :



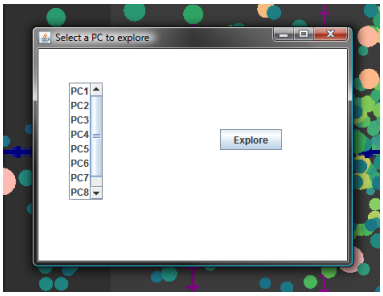
Menu Explore



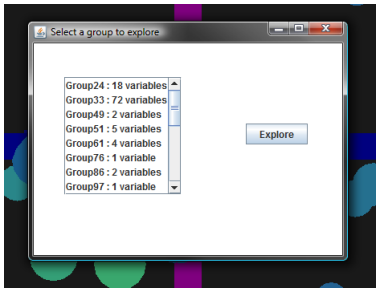
The explore menu contains all the functions that allows us to explore the data or, more precisely, the data represented by the principal components or the groups.

To understand how we can explore the data, we need to recall that the original attributes have been gathered in groups according to their means and their coefficients of variation. The groups have been transformed into interval data and we have performed principal component analysis on these interval variables. These principal components are shown on the basic multiple scatter plot. To explore the data, more precisely, the attributes, we have three options : select one principal component, select one group or select some attributes.

The **select a principal component** option means that we are going to focus on the groups that have an important influence on this principal component. Selecting a principal component thus means that we are going to make the representation of the attributes that are contained into those groups. If the number of groups is too big, we apply the principal component analysis a second time and the results that are shown are the results of the principal component analysis. If the number of groups is not too big, we will show the groups directly.



The **select a group** option means that we are going to focus on the attributes contained in that group. If the number of elements is too big, we apply the construction method a second time (with smaller percentage of variation for the mean and the coefficient of variation). Again, if the number of groups is too big, we apply the principal component analysis.



The **select some variables** option means that we are going to focus on some particular attributes. These attributes will be represented by a scatter plot or by a multiple scatter plot if the number of attributes selected is greater than 2.